Parameterizing the Permanent: Genus, Apices, Minors, Evaluation mod 2^k

Radu Curticapean*, Mingji Xia†

November 10, 2015

Abstract

We identify and study relevant structural parameters for the problem PerfMatch of counting perfect matchings in a given input graph G. These generalize the well-known tractable planar case, and they include the *genus* of G, its *apex number* (the minimum number of vertices whose removal renders G planar), and its *Hadwiger number* (the size of a largest clique minor).

To study these parameters, we first introduce the notion of *combined matchgates*, a general technique that bridges parameterized counting problems and the theory of so-called Holants and matchgates: Using combined matchgates, we can simulate certain non-existing gadgets F as linear combinations of $t = \mathcal{O}(1)$ existing gadgets. If a graph G features k occurrences of F, we can then reduce G to t^k graphs that feature only existing gadgets, thus enabling parameterized reductions.

As applications of this technique, we simplify known $4^g n^{O(1)}$ time algorithms for PerfMatch on graphs of genus g. Orthogonally to this, we show $\#\mathsf{W}[1]$-hardness of the permanent on k-apex graphs, implying its $\#\mathsf{W}[1]$-hardness under the Hadwiger number. Additionally, we rule out $n^{o(k/\log k)}$ time algorithms under the counting exponential-time hypothesis $\#\mathsf{ETH}$.

Finally, we use combined matchgates to prove $\oplus\mathsf{W}[1]$-hardness of evaluating the permanent modulo 2^k, complementing an $\mathcal{O}(n^{4k-3})$ time algorithm by Valiant and answering an open question of Björklund. We also obtain a lower bound of $n^{\Omega(k/\log k)}$ under the parity version $\oplus\mathsf{ETH}$ of the exponential-time hypothesis.

*Simons Institute for the Theory of Computing, Berkeley, USA, and Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary. Supported by ERC Starting Grant PARAMTIGHT, No. 280152.

†Institute of Software, Chinese Academy of Sciences, Beijing, China. Supported by China National 973 program 2014CB340301, China Basic Research Program (973) Grant 2014CB340302, NSFC 61003030 and NSFC 61170073.

Contents

1 Introduction **2**
- 1.1 Genus, apices and excluded minors . . . 3
- 1.2 Evaluating the permanent modulo 2^k . . . 5
- 1.3 Proof technique: Linear combinations of signatures . . . 6

2 General preliminaries **6**
- 2.1 Parameterized complexity . . . 7
- 2.2 Exponential-time complexity . . . 7
- 2.3 Grid tilings and vertex-colored subgraphs . . . 8

3 Holants, matchgates, linear combinations of signatures **10**
- 3.1 Signature graphs and Holants . . . 10
- 3.2 Gates and matchgates . . . 11
- 3.3 Linear combinations of matchgate signatures . . . 14

4 PerfMatch on bounded-genus graphs **15**
- 4.1 The algorithm . . . 15
- 4.2 Additional remarks . . . 17

5 The permanent on k-apex graphs **17**
- 5.1 Global construction . . . 18
- 5.2 Realizing cell signatures . . . 20
- 5.3 Calculating the signatures of Φ and Φ' . . . 21

6 The permanent modulo 2^k **23**
- 6.1 The main idea . . . 24
- 6.2 Revisiting the cell gate . . . 24
- 6.3 Linear combinations via discrete derivatives . . . 25
- 6.4 Calculating the signature of Γ . . . 28

1 Introduction

The study of counting problems has become a classical subfield of computational complexity since Valiant's seminal papers [51, 52] introduced the class #P and established #P-hardness of counting perfect matchings in bipartite graphs. In particular, this proves #P-hardness of the following generalized problem: Given a graph G with edge-weights $w : E(G) \to \mathbb{Q}$, compute the quantity

$$\mathrm{PerfMatch}(G) := \sum_{\substack{M \subseteq E(G) \\ \text{perfect matching of } G}} \prod_{e \in M} w(e).$$

In statistical physics, PerfMatch is known as the *partition function* of the *dimer model* [48, 35, 36], and the first nontrivial algorithms for the evaluation of this quantity stem from this area. This includes the celebrated *FKT method*, a polynomial-time algorithm for computing PerfMatch on planar graphs [36]. Roughly speaking, this algorithm proceeds as follows: Given a planar graph G, it constructs a *Pfaffian orientation* F of G, which we may view as a subset $F \subseteq E(G)$ with the following miraculous property: If we define a matrix A from the adjacency matrix of G by flipping the signs of edges in F, then $(\mathrm{PerfMatch}(G))^2 = \det(A)$. Overall, this yields a reduction from planar PerfMatch to the determinant.

In algebra and combinatorics, the quantity PerfMatch(G) for a bipartite graph G with $n+n$ vertices is better known as the *permanent* of the biadjacency matrix A of G, defined by

$$\mathrm{perm}(A) = \sum_{\substack{\sigma:[n]\to[n] \\ \text{is a permutation}}} \prod_{i=1}^{n} A_{i,\sigma(i)}.$$

The permanent is central to algebraic complexity theory, which aims at proving the permanent to be inherently harder than the similar-looking determinant [1, 43, 4]. This would imply an algebraic analogue of $\mathsf{P} \neq \mathsf{NP}$ [50].

In order to obtain a more refined view on the complexity of the permanent, and to cope with its hardness in view of practical applications, various relaxations of this problem were studied: A celebrated randomized **approximation** scheme [34, 33] allows one to approximate the permanent on matrices with non-negative entries. Furthermore, on some **restricted graph classes**, PerfMatch can be solved in time $\mathcal{O}(n^3)$: This includes the above-mentioned planar graphs, and in fact, all graph classes of bounded genus [29, 49, 44]. (We will present more classes in the remainder of the introduction.) As another relaxation, **modular evaluation** of the permanent was studied in Valiant's original paper [51]: He showed that the permanent modulo $m = 2^k$ can be computed in time $n^{\mathcal{O}(k)}$ for all $k \in \mathbb{N}$, but for all m containing an odd prime factor, the evaluation modulo m is NP-hard under randomized reductions.

In this paper, we consider another such refinement (and generalize existing ones) by investigating the permanent in the framework of **parameterized complexity**. This area was initiated by Downey and Fellows [24, 25] and was adapted to counting problems by Flum and Grohe [26] and McCartin [42]. In parameterized counting complexity, the objects in study are counting problems that come with *parameterizations* $\pi : \{0,1\}^* \to \mathbb{N}$, and a central question is whether such problems are *fixed-parameter tractable (fpt)*. A given problem is fpt if it can be solved in time $f(\pi(x))|x|^{\mathcal{O}(1)}$ on input x, for a computable function f that depends only on the parameter value, but not on $|x|$. If we fail to find an fpt-algorithm for a given parameterized problem, we can often give evidence that no such algorithm exists by proving its $\#\mathsf{W}[1]$-hardness, the parameterized analogue of $\#\mathsf{P}$-hardness. (A more detailed exposition can be found in Section 2.)

By studying natural parameterizations π of the input, we obtain a fine-grained complexity analysis that could not be achieved by considering the input size $|x|$ alone. For instance, consider the problem VertexCover, which asks whether a graph G on n vertices admits a vertex-cover of size k. This problem is NP-complete, but it can be solved in time $n^{\mathcal{O}(k)}$ for every fixed k, and it is actually even fpt in the parameter k, as we can find [24] and even count [27] vertex-covers of size k in time $2^k n^{\mathcal{O}(1)}$. On the other hand, we can decide in polynomial time whether G contains a matching of size k, but the problem of counting k-matchings is $\#\mathsf{P}$-complete, and in fact even $\#\mathsf{W}[1]$-complete when parameterized by k [13, 16].

1.1 Genus, apices and excluded minors

To investigate the parameterized complexity of the permanent, we first identify interesting parameterizations for this problem. For instance, the maximum degree $\Delta(G)$ of the input graph G is not particularly interesting, since the permanent is already $\#\mathsf{P}$-complete on 3-regular graphs [17]. That is, even an $n^{f(\Delta(G))}$ time algorithm for some function f (and an fpt-algorithm in particular) would imply $\mathsf{P} = \#\mathsf{P}$. However, it turns out that the known polynomial-time solvable graph classes for PerfMatch point us towards a natural parameter, namely the size of a smallest excluded minor. Here, a minor H of a graph G is a graph that can be obtained from G by deletions of edges and/or vertices, and contraction of edges. To explain the significance of minors for counting perfect matchings, we first survey the known algorithms for PerfMatch, all of which can be considered as generalizations of the FKT method for planar graphs.

Excluding $K_{3,3}$ or K_5: It was shown by Little [37] and later by Vazirani [55] (who gave a parallelized algorithm) that PerfMatch can be solved in time $\mathcal{O}(n^3)$ on graphs excluding the minor $K_{3,3}$. A similar result was recently shown by Straub et al. [47] for graphs excluding K_5. Note that the FKT method gives an $\mathcal{O}(n^3)$ time algorithm on graphs excluding *both* $K_{3,3}$ and K_5, whereas the two above algorithms

show that excluding *either* minor entails the polynomial-time solvability of PerfMatch. For the $K_{3,3}$-free case, this was shown by constructing a Pfaffian orientation. The K_5-free case was shown by a different technique; in particular, K_5-free graphs do not necessarily admit Pfaffian orientations.

Excluding single-crossing minors: Extending the above item, it was recently shown by Curticapean [14] that PerfMatch can be solved in time $\mathcal{O}(n^4)$ on any class excluding a fixed *single-crossing minor* H, i.e., a minor that can be drawn in the plane with at most one crossing, such as $K_{3,3}$ or K_5. In fact, it is shown that PerfMatch is fpt in the size of the smallest excluded single-crossing minor. This algorithm does not inherently rely upon Pfaffian orientations, apart from a black-box algorithm for planar PerfMatch.

Bounded-genus graphs: Another line of extensions of the FKT method is to graphs of bounded *genus*: It was shown independently by Gallucio and Loebl [29], Tesler [49] and Regge and Zechina [44] that PerfMatch can be solved in time $\mathcal{O}(4^g n^3)$ on n-vertex graphs G of genus g. In the framework of fixed-parameter tractability, this can be read as PerfMatch being fpt when parameterized by the genus of G. The algorithms for the bounded-genus case proceed by expressing PerfMatch(G) as the linear combination of 4^g determinants derived from Pfaffian orientations. In the present paper, we give an alternative proof of this theorem that proceeds by reduction to 4^g instances of planar PerfMatch. Together with the previous item, this eliminates the need for Pfaffian orientations from all known algorithms for PerfMatch except for the planar case.

From the above list, we can draw the conclusion that every *known* polynomial-time solvable graph class for PerfMatch excludes some fixed minor.[1] This is clear for the first two items, and furthermore, the graphs of genus $g \in \mathbb{N}$ are easily seen to exclude a complete graph of size $\mathcal{O}(g)$. Since this shows that excluded minors have been a driving force behind polynomial-time algorithms for PerfMatch, it is natural to study this problem under the more general *Hadwiger number*

$$\mathrm{hadw}(G) := \max\{k \in \mathbb{N} : \ G \text{ contains a } K_k\text{-minor}\}.$$

Note that planar graphs have Hadwiger number at most 4. More generally, if the genus of G or the size of the smallest excluded single-crossing minor is bounded, then $\mathrm{hadw}(G)$ is bounded as well, but the converse does not hold. However, the *Graph Structure Theorem* [45], a celebrated result in graph minor theory [46], yields a decomposition of the graphs with fixed Hadwiger number k into graphs that have genus $c = c(k)$ except for c occurrences of certain defects, namely so-called vortices and apices. Such decompositions have proven immensely useful for fpt-algorithms on graphs excluding fixed minors, see [40, 22, 21, 20, 19, 28]. If a problem can be solved efficiently on planar instances and we can extend this to bounded-genus instances, as in the case of PerfMatch, then with a leap of faith, the Graph Structure Theorem allows us to hope for an fpt-algorithm under the more general parameterization by Hadwiger number. Our following negative result however shatters these hopes for the case of PerfMatch.

Theorem 1.1. *The zero-one permanent is* #W[1]*-hard when parameterized by the Hadwiger number. In other words, computing* PerfMatch *is* #W[1]*-hard when parameterized by the Hadwiger number, even on bipartite graphs without edge-weights.*

We show this by proving the following stronger statement: Let us define the apex number

$$\mathrm{apex}(G) := \min\{k \in \mathbb{N} \mid \exists S \subseteq V(G) \text{ of size } k : \ G - S \text{ is planar}\}.$$

This parameter, studied in [41], measures the distance of a graph to planarity by vertex deletions. Note that planar graphs have apex number 0. Using the apex number as parameter, we can generalize planar

[1]This statement comes with a caveat: For instance, we can determine the number of perfect matchings in a complete graph in polynomial time by means of a closed formula. The class of complete graphs clearly excludes no fixed minor. However, we cannot solve the (weighted) problem PerfMatch on this class in polynomial time, as edge-weights would allow us to simulate arbitrary graphs, for which counting perfect matchings is #P-complete.

graphs in a way that is orthogonal to the genus parameter: There are graphs on which any one of these parameters is bounded, while the other is not. However, it can be verified that $\mathrm{hadw}(G) \leq \mathcal{O}(\mathrm{apex}(G))$. This allows us to obtain Theorem 1.1 as a corollary from the following result, which we consider to be of independent interest.

Theorem 1.2. *The permanent is* $\#\mathsf{W}[1]$*-hard when parameterized by the apex number. Assuming the exponential-time hypothesis* $\#\mathsf{ETH}$*, it admits no* $n^{o(k/\log k)}$ *time algorithm on* k*-apex graphs with* n *vertices.*

This contrasts with the fpt-algorithm for PerfMatch when parameterized by genus. We observe that PerfMatch can be computed easily in time $n^{k+\mathcal{O}(1)}$ on k-apex graphs by means of brute-force, so the lower bound under $\#\mathsf{ETH}$ is almost tight. However, it should be noted that no similar algorithm is known for the Hadwiger number: At least to us, it remains an important open question whether PerfMatch can be solved in time $n^{f(k)}$ on graphs excluding the complete graph K_k as minor.

1.2 Evaluating the permanent modulo 2^k

In the following, we depart from structural parameters of the input graph G and consider the evaluation of the permanent modulo 2^k. In the seminal paper [51], not only did Valiant prove $\#\mathsf{P}$-completeness of the permanent, but he also studied the complexity of evaluating the permanent modulo fixed numbers $m \in \mathbb{N}$.

Observe that $\mathrm{perm}(A)$ and $\det(A)$ are equivalent modulo 2 for any matrix A, giving a polynomial-time algorithm for the permanent modulo 2. On the other hand, for odd primes p, Valiant's original proof shows that the permanent modulo p is $\mathsf{Mod}_p\mathsf{P}$-complete. That is, we can reduce counting satisfying assignments to 3-CNF formulas modulo p to the permanent modulo p. This also shows the NP-hardness of the latter problem under randomized reductions, and this holds more generally whenever the modulus m contains an odd prime factor, that is, when m is not a power of two.

For the remaining cases $m = 2^k$ however, Valiant [51] showed an $\mathcal{O}(n^{4k})$ time algorithm for evaluating the permanent modulo 2^k on n-vertex graphs, which was recently improved to $n^{k+\mathcal{O}(1)}$ time by Björklund, Husfeldt and Lyckberg [3]. Given these results, it is natural to study this problem in the framework of parameterized complexity, thus asking whether we can compute the permanent modulo 2^k in time $n^{o(k)}$ or even $f(k)n^{\mathcal{O}(1)}$. This was also posed as an open problem in [3]. Please recall that this question is indeed only interesting for $m = 2^k$: As stated in the previous paragraph, on all other *fixed* $m \in \mathbb{N}$, the problem is NP-hard under randomized reductions.

We rule out the fixed-parameter tractability of the permanent modulo 2^k by the following stronger hardness result, which also establishes an unexpected connection to the apex parameter introduced before: Evaluating the permanent modulo 2^k on k-apex graphs is $\oplus\mathsf{W}[1]$-hard, that is, an fpt-algorithm for this problem would imply one for counting k-cliques modulo 2, a problem that was shown to be $\mathsf{W}[1]$-hard under randomized reductions by a recent result of Björkund, Dell and Husfeldt [2]. We also obtain an almost-tight lower bound under $\oplus\mathsf{ETH}$, the parity version of the exponential-time hypothesis ETH.

Theorem 1.3. *The evaluation of the permanent modulo* 2^k *is* $\oplus\mathsf{W}[1]$*-hard when parameterized by* k*, even when restricted to* k*-apex graphs. Assuming* $\oplus\mathsf{ETH}$*, there is no* $n^{o(k/\log k)}$ *time algorithm for this problem.*

Theorem 1.3 is proven by reduction from the following problem $\oplus$PartitionedSub: Given vertex-colored graphs H and G as input, where each color in H appears exactly once, count modulo 2 the subgraphs of G that are isomorphic to H, respecting colors. It was shown that the decision version of this problem, which is $\mathsf{W}[1]$-hard, can be reduced to $\oplus$PartitionedSub by means of randomized reductions [2]. Furthermore, assuming $\oplus\mathsf{ETH}$, an argument by Marx [38] implies that $\oplus$PartitionedSub cannot be solved in time $n^{o(\ell/\log \ell)}$ for ℓ-edge graphs H and n-vertex graphs G.

In our reduction, we transform a given instance (H, G) for $\oplus$PartitionedSub with an ℓ-edge graph H to 3^ℓ instances of the permanent modulo $2^{2\ell+1}$ on 2ℓ-apex graphs with $\mathcal{O}(\ell^2 n^2)$ vertices. Thus, if we can prove better lower bounds for finding k-edge subgraphs, then those bounds carry over to the seemingly unrelated problem of evaluating permanents modulo 2^k, even on k-apex graphs. On the other hand, a randomized $n^{o(k)}$ time algorithm for the permanent modulo 2^k on k-apex graphs would imply one for PartitionedSub on k-edge graphs H, thus falsifying a hypothesis posed by Marx [38].

1.3 Proof technique: Linear combinations of signatures

We phrase our proofs in the language of so-called Holant problems [8] and matchgates [8, 5, 6]. Please consider Section 3 for a more detailed introduction into these topics. In our proofs, we reformulate parameterized counting problems as Holant problems (specific weighted sums over assignments to the edges of graphs) and then try to realize the occurring signatures (local constraints at vertices) by certain matchgates (gadgets). However, many required signatures cannot be realized by matchgates. The key idea in this paper is that such unrealizable signatures can sometimes still be realized as *linear combinations* of matchgate signatures.

To this end, we proceed as follows: First, we show how to simulate non-existing gadgets F as a formal linear combination of realizable gadgets $F_1, \ldots, F_t$, typically with $t = \mathcal{O}(1)$. Then, if a graph G features k occurrences of F, we can easily reduce G to t^k graphs that feature only occurrences of $F_1, \ldots, F_t$. Each of these t^k graphs can then be handled by an algorithm (when we aim at positive results) or by an oracle call (when proving hardness results). The generality of our technique allows it to be applied to various parameterized problems. For instance, a recent #W[1]-hardness proof for counting k-matchings [16] can also be rephrased in this framework.

As pointed out by Tyson Williams, a similar idea appears under the notion of *vanishing signatures* [30, 7]. These however apply linear combinations in a quite different setting. In particular, they consider no connections to parameterized complexity.

Organization of the paper

In Section 2, we introduce notions from parameterized complexity, exponential-time complexity, and we prove #W[1]-hardness of a modified version of the problem #GridTiling, our main reduction source for subsequent hardness proofs. In Section 3, we introduce Holant problems and matchgates, including some particular matchgates required in later sections. We also introduce our proof technique of linearly combined signatures. This finishes the general introduction of our proof techniques.

In Section 4, we then give a first application of the machinery developed in the previous sections by proving a $4^g \cdot n^{\mathcal{O}(1)}$ time algorithm for PerfMatch on graphs of genus g. In Section 5, we then prove Theorem 1.2, which asserts #W[1]-hardness of PerfMatch on bipartite unweighted k-apex graphs and implies Theorem 1.1, the hardness under the Hadwiger number parameter. In Section 6, we introduce a more involved construction and an additional technique called *discrete derivatives* to transform the argument from Section 5 to a proof of Theorem 1.3.

2 General preliminaries

For $n \in \mathbb{N}$, we write $[n] := \{1, \ldots, n\}$. The graphs G in this paper are undirected, but they may feature parallel edges and edge-weights. All *hardness results* are however shown for *simple* graphs featuring no parallel edges and no edge-weights. We write $uv \in E(G)$ for an edge of G, and given $v \in V(G)$, we denote the edges incident with v by $I(v)$. Sometimes, we consider graphs to be embedded on surfaces, see [23].

For numbers $n \in \mathbb{N}$, we abbreviate $\oplus n := (n \bmod 2)$. Given a bitstring $x \in \{0,1\}^*$, we write $\mathrm{hw}(x) := \sum_i x_i$ for its *Hamming weight*, and we define

$$\begin{aligned} \texttt{ODD}(x) &:= \oplus \mathrm{hw}(x), \\ \texttt{EVEN}(x) &:= 1 - \oplus \mathrm{hw}(x). \end{aligned}$$

We write $\mathrm{supp}(f)$ for the support of a function f. For predicates φ, we use the Iverson bracket notation

$$[\varphi] := \begin{cases} 1 & \varphi \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

Let A and B be sets; we define certain abbreviations for subsets of $A \times B$. For $b \in B$, we write $(\star, b) = \{(a, b) \mid a \in A\}$ for the *column* at b. For $a \in A$, we write $(a, \star) = \{(a, b) \mid b \in B\}$ for the *row* at

a. We use this notation only when A and B are unambiguous from the context. For $k \in \mathbb{N}$, we say that $(i,j) \in [k]^2$ and $(i',j') \in [k]^2$ are *vertically adjacent* if $|i-i'| = 1$ and $j = j'$. Likewise, we call such pairs *horizontally adjacent* if $|j-j'| = 1$ and $i = i'$.

2.1 Parameterized complexity

Parameterized counting problems are problems A/π, where $\mathrm{A} : \{0,1\}^* \to \mathbb{C}$ is a counting problem and $\pi : \{0,1\}^* \to \mathbb{N}$ is a polynomial-time computable parameterization, see [26]. We define FPT as the class of all problems A/π such that A can be solved in time $f(\pi(x))|x|^{\mathcal{O}(1)}$. Likewise, we define XP as the class of problems A/π that can be solved in time $|x|^{f(\pi(x))}$, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function. In the following, we define the classes $\mathsf{W}[1]$, $\#\mathsf{W}[1]$ and $\oplus\mathsf{W}[1]$ we referred to in the introduction, using the following reduction notions.

Definition 2.1 ([26])**.** Let A/π and B/π' be parameterized counting problems.

- We call $f : \{0,1\}^* \to \{0,1\}^*$ a *parsimonious fpt-reduction* and write $\mathrm{A}/\pi \leq_{\mathit{fpt}}^{\mathit{pars}} \mathrm{B}/\pi'$ if there are computable functions r, s such that the following holds for all $x \in \{0,1\}^*$:

  1. We have $\mathrm{A}(x) = \mathrm{B}(f(x))$.
  2. The running time of f is bounded by $r(\pi(x)) \cdot |x|^{\mathcal{O}(1)}$.
  3. We have $\pi'(f(x)) \leq s(\pi(x))$.

  If A and B are decision problems, replace the first condition by "$x \in \mathrm{A}$ iff $f(x) \in \mathrm{B}$", and write $\mathrm{A}/\pi \leq_{\mathit{fpt}} \mathrm{B}/\pi'$.

- We call an algorithm $\mathbb{T}$ a *Turing fpt-reduction* and write $\mathrm{A}/\pi \leq_{\mathit{fpt}}^{T} \mathrm{B}/\pi'$ if there are computable functions r and s such that the following holds for all $x \in \{0,1\}^*$: Firstly, the running time of $\mathbb{T}$ on x is bounded by $r(\pi(x))|x|^{\mathcal{O}(1)}$. Secondly, every oracle query y issued by $\mathbb{T}$ on x satisfies $\pi'(y) \leq s(\pi(x))$.

We introduce $\mathsf{W}[1]$, $\oplus\mathsf{W}[1]$ and $\#\mathsf{W}[1]$ as the closures of clique-related problems under fpt-reductions.

Definition 2.2. Consider the following parameterized problems and complexity classes:

- Let Clique/k denote the problem of *deciding*, on input a graph G and $k \in \mathbb{N}$, whether G contains a k-clique. Let $\mathsf{W}[1]$ denote the set of all problems A/π with $\mathrm{A}/\pi \leq_{\mathit{fpt}} \mathrm{Clique}/k$.

- Let $\#\mathrm{Clique}/k$ denote the problem of determining, on input G and k, the *number* of k-cliques in G. Let $\#\mathsf{W}[1]$ denote the set of all problems A/π with $\mathrm{A}/\pi \leq_{\mathit{fpt}}^{\mathit{pars}} \#\mathrm{Clique}/k$.

- Let $\oplus\mathrm{Clique}/k$ denote the problem of *deciding*, on input G and k, whether G contains an *odd* number of k-cliques. Let $\oplus\mathsf{W}[1]$ denote the set of all A/π with $\mathrm{A}/\pi \leq_{\mathit{fpt}} \oplus\mathrm{Clique}/k$.

It is a standard assumption of parameterized complexity theory that $\mathsf{FPT} \neq \mathsf{W}[1]$ holds, implying $\mathsf{FPT} \neq \#\mathsf{W}[1]$. The problem Clique/k is $\mathsf{W}[1]$-complete by definition, so this assumption can equivalently be considered as the statement that Clique/k is not fixed-parameter tractable. Furthermore, it has been recently shown in [2, Theorem 5] that $\oplus\mathrm{Clique}/k$ is $\mathsf{W}[1]$-hard under randomized parameterized reductions with constant one-sided error. Therefore, an fpt-algorithm for $\oplus\mathrm{Clique}/k$ would imply a randomized fpt-algorithm for Clique/k, which is considered almost as unlikely as $\mathsf{FPT} = \mathsf{W}[1]$.

2.2 Exponential-time complexity

We also consider conditional lower bounds on the running times required to solve problems. These are based on different exponential-time hypotheses, introduced by [31, 32] and [18].

Definition 2.3. The exponential-time hypothesis ETH, introduced in [31, 32], claims that the satisfiability of 3-CNF formulas on n variables cannot be decided in time $2^{o(n)}n^{O(1)}$. The hypothesis $\#\mathsf{ETH}$ postulates the same lower bound for counting the number of satisfying assignments to 3-CNF formulas, and $\oplus\mathsf{ETH}$ postulates the same for computing the parity of the number of satisfying assignments.

The hypothesis ETH implies a lower bound for Clique/k, and thus also $\mathsf{FPT} \neq \mathsf{W}[1]$: It was shown in [11, 12] that Clique/k cannot be solved in time $f(k) \cdot n^{o(k)}$ on n-vertex graphs, for any computable function f. Furthermore, if a problem A/π cannot be solved in time $f(k) \cdot n^{g(k)}$ under ETH, and we can reduce A/π to B/π' with a reduction f that satisfies $\pi'(f(x)) \in \mathcal{O}(\pi(x))$ for all x, then it can be seen that B/π' cannot be solved in time $f'(k) \cdot n^{\Omega(g(k))}$ under ETH, for any computable function f'.

By an isolation argument similar to the Valiant-Vazirani theorem [54], it was shown in [10] that a $2^{o(n)}$ time algorithm for counting satisfying assignments to 3-CNF formulas modulo 2 implies a randomized $2^{o(n)}$ time algorithm for deciding the existence of a satisfying assignment. In other words, a randomized version rETH of ETH implies $\oplus\mathsf{ETH}$; see also [18] for more details.

2.3 Grid tilings and vertex-colored subgraphs

We will reduce from the problem GridTiling of deciding the existence of grid tilings, as well as its counting version #GridTiling and its parity counting version $\oplus$GridTiling. The decision version GridTiling was introduced by Marx [39] in order to obtain lower bounds for planar multiway cut, but grid tilings have since proven to be generally useful for proving hardness of problems on planar structures [40].

Definition 2.4. The inputs to the problem GridTiling are numbers $n, k \in \mathbb{N}$, together with a set $\mathcal{C} \subseteq [k]^2$ and a function $\mathcal{T} : \mathcal{C} \to 2^{[n]^2}$. The task is to decide whether there exists a *grid tiling* of $\mathcal{T}$, i.e., a function $a : [k]^2 \to [n]^2$ such that:

1. For horizontally adjacent $\kappa, \kappa' \in [k]^2$, the first components of $a(\kappa)$ and $a(\kappa')$ agree.
2. For vertically adjacent $\kappa, \kappa' \in [k]^2$, the second components of $a(\kappa)$ and $a(\kappa')$ agree.
3. For all $\kappa \in \mathcal{C}$, we have $a(\kappa) \in \mathcal{T}(\kappa)$.

On the same inputs, we also define the problem #GridTiling, which asks to determine the *number* of grid tilings, and the problem $\oplus$GridTiling, which asks to determine the *parity* of this number. All three problems are parameterized by k.

It should be noted that our definition of GridTiling is actually a generalized version of Marx's original formulation [39]: In his definition, the set $\mathcal{C}$ of any instance is fixed to $\mathcal{C} = [k]^2$. That is, the third condition of Definition 2.4 is required to apply for *all* $\kappa \in [k]^2$, whereas in our formulation, only a subset is relevant. In particular, we may choose sparse subsets $\mathcal{C}$ with $|\mathcal{C}| = \mathcal{O}(k)$, which will make the generalized grid tiling problems very useful in proving lower bounds under the exponential-time hypotheses.

By reduction from k-cliques, Marx showed that GridTiling is complete for $\mathsf{W}[1]$. A simple adaptation of this reduction shows that the same holds for its counting and parity version, where $\#\mathsf{W}[1]$ and $\oplus\mathsf{W}[1]$ take the part of $\mathsf{W}[1]$. In the remainder of this subsection, we give a different reduction, which chooses partitioned subgraph isomorphisms rather than k-cliques as a reduction source. This allows us to transfer an almost-tight conditional lower bound for subgraph isomorphisms under ETH to GridTiling.

Definition 2.5. For $k \in \mathbb{N}$, a $[k]$-*colored graph* is a pair (H, c), where H is a graph and $c : V(H) \to [k]$ is a coloring. We call (H, c) *colorful* if c is bijective. This implies of course that H has k vertices.

For $[k]$-colored graphs (H, c) and (G, c'), we say that (H, c) is *color-preserving isomorphic* to (G, c') if there exists an isomorphism f from H to G such that $c(v) = c'(f(v))$ holds for all $v \in V(H)$. To simplify notation, we will often write G rather than (G, c) for a colored graph.

The problem PartitionedSub is defined as follows: The input consists of $[k]$-colored graphs H and G, where H is colorful. The task is to decide whether there exists a copy of H in G, which is a (not necessarily induced) subgraph F of G such that H and F are color-preserving isomorphic. Likewise, the problem

#PartitionedSub asks to determine the *number* of copies of H in G, and $\oplus$PartitionedSub asks to determine its parity. All problems are parameterized by k.

It can be shown by a parsimonious reduction from Clique that the problem PartitionedSub is $\mathsf{W}[1]$-complete, and this implies similar statements for its other variants as well. We omit the elementary proof.

Lemma 2.6. *The three variants of* PartitionedSub *are complete for* $\mathsf{W}[1]$, $\#\mathsf{W}[1]$ *or* $\oplus\mathsf{W}[1]$*, respectively.*

Remark 2.7. Let H and G be $[k]$-colored such that H is colorful; we assume $V(H) = [k]$ without limitation of generality. If F is a H-copy in G and $uv \in E(F)$ is an edge with endpoints of colors i and j for some $i, j \in [k]$, then the edge ij must be present in H.

We may therefore assume the following: Whenever an instance (H, G) to PartitionedSub is given, then for all $i, j \in [k]$ with $ij \notin E(H)$, the graph G contains no edges between i-colored and j-colored vertices. Otherwise, we can delete these edges without affecting the set of color-preserving H-copies in G.

In the following, we consider instances (H, G) for PartitionedSub with $n = |V(G)|$ and $k = |V(H)|$. We can solve each such instance in time $n^{\mathcal{O}(k)}$ by brute force, and by reduction from Clique, it was shown that algorithms with running time $f(k) \cdot n^{o(k)}$ would refute ETH, see [11, 12].

This lower bound alone would however not suffice for our purposes of proving tight lower bounds: In the reductions from PartitionedSub to the permanent we construct later, each *edge* of H will incur some constant parameter blowup. As an example, on instances (H, G), our reduction images for the permanent will have $\mathcal{O}(|E(H)|)$ apices, which amounts to $\mathcal{O}(k^2)$ apices if H is a k-clique. Thus, if we reduced from Clique for our lower bound, then ETH could only rule out algorithms with running time $n^{o(\sqrt{t})}$ for the permanent on t-apex graphs. This is however obviously far from the upper bound of $\mathcal{O}(n^{t+3})$ time obtained by the brute-force algorithm, and we would not consider such a result to be satisfactory.

To avoid this problem, we use a refined lower bound for PartitionedSub, shown also by Marx, which allows to assume that H has constant degree, and thus, only $\mathcal{O}(k)$ edges, see [38, Corollary 6.3].

Theorem 2.8 ([38])**.** *Assuming* ETH*, there is a universal constant* C^* *such that* PartitionedSub *cannot be solved in time* $f(k) \cdot n^{o(k/\log k)}$*, for any computable function* f*, even on instances* (H, G) *where* H *has maximum degree at most* C^**. The same applies to the variants* #PartitionedSub *and* $\oplus$PartitionedSub*, assuming* $\#\mathsf{ETH}$ *and* $\oplus\mathsf{ETH}$ *respectively.*

Using Lemma 2.6 and Theorem 2.8, we can then prove similar lower bounds for grid tilings.

Theorem 2.9. *The three variants of* GridTiling *are complete for* $\mathsf{W}[1]$, $\#\mathsf{W}[1]$ *and* $\oplus\mathsf{W}[1]$*, respectively. Furthermore, the problems admit no* $n^{o(k/\log k)}$ *time algorithms, even on instances with* $|\mathcal{C}| = \mathcal{O}(k)$*, unless* ETH*,* $\#\mathsf{ETH}$ *or* $\oplus\mathsf{ETH}$ *fails, respectively.*

Proof. Let G and H be $[k]$-vertex-colored, where we assume $V(G) = [n]$ and $V(H) = [k]$. Replace each edge uv in G by the directed edges uv and vu, then add all self-loops to G to obtain a colored directed graph G'. Define the colorful directed graph H' by applying the same operations on H. Then we can observe that the color-preserving H-copies in G stand in bijection with the color-preserving H'-copies in G'.

For $i, j \in [k]$, write $E_{i,j} = E_{i,j}(G')$ for the set of directed edges in G' from i-colored vertices to j-colored vertices. By Remark 2.7, we may assume that $E_{i,j} = \emptyset$ if $ij \notin E(H')$. Note that $E_{i,j} \subseteq [n]^2$; we use this to define an instance $(n, k, \mathcal{C}, \mathcal{T})$ for GridTiling by declaring $\mathcal{C} := E(H')$ and $\mathcal{T}(i, j) := E_{i,j}$ for all $ij \in E(H')$. We then claim that the grid tilings of this instance correspond bijectively to the H'-copies in G'. This gives a parsimonious reduction from #PartitionedSub to #GridTiling, which, together with Lemma 2.6 and Theorem 2.8, implies all claims of the theorem.

It remains to verify the claimed bijection: The third property of Definition 2.4 implies that every tiling $a : [k]^2 \to [n]^2$ encodes an edge-subset $S_a \subseteq E(G')$ with $|S_a| = |E(H)|$ that picks exactly one element from $E_{i,j}$ for each $ij \in E(H')$. If the edges in S_a are incident with exactly k distinct vertices, then S_a induces a H'-copy in G'. By the first two properties of Definition 2.4, the edge set S_a contains exactly k distinct endpoints and k distinct starting points. Since $E_{i,i}$ for $i \in [k]$ contains only self-loops, the sets of endpoints and starting points of edges in S_a are identical, which implies that S_a is a H'-copy in G'. Conversely, every H'-copy in G' can be mapped to such a grid tiling by reversing this operation. $\square$

In the following, we add a small technical extension to Theorem 2.9 that allows us to assume each input instance to be balanced along rows or columns in a certain way. While it is almost trivial to ensure this balance property by adding dummy elements, it turns out to be very useful in our reductions from GridTiling.

Lemma 2.10. *Let $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ be an instance for* GridTiling *and let $\mathfrak{W}$ be either of the words "horizontal" or "vertical". In polynomial time, we can then compute a number $T \in \mathbb{N}$ and a grid tiling instance $\mathcal{A}' = (n', k, \mathcal{C}, \mathcal{T}')$ with $n' = \mathcal{O}(k^2 n)$ such that:*

1. *The instances $\mathcal{A}$ and $\mathcal{A}'$ have precisely the same grid tilings.*
2. *For $u \in [n]$, write $(u, \star) := \{(u, v) \mid v \in [n]\}$. For $v \in [n]$, write $(\star, v) := \{(u, v) \mid u \in [n]\}$.*
   (a) *If $\mathfrak{W}$ is "horizontal", then for all $\kappa \in \mathcal{C}$ and $u \in [n']$, we have $|\mathcal{T}'(\kappa) \cap (u, \star)| = T$.*
   (b) *If $\mathfrak{W}$ is "vertical", then for all $\kappa \in \mathcal{C}$ and $v \in [n']$, we have $|\mathcal{T}'(\kappa) \cap (\star, v)| = T$.*

Proof. We show the statement if $\mathfrak{W}$ is "vertical"; the horizontal case is shown in exactly the same manner. Let us first define

$$T_{\kappa,v} := |\mathcal{T}(\kappa) \cap (\star, v)| \quad \text{for } \kappa \in [k]^2 \text{ and } v \in [n],$$

that is, the number of elements in the v-th column of $\mathcal{T}(\kappa)$. Then we define

$$T := \max_{\kappa \in [k]^2,\, v \in [n]} T_{\kappa,v}$$

and let $n' := n + k^2 T$. Consider $[n']$ to be partitioned into $[n]$ and k^2 consecutive "dummy" blocks B_κ for $\kappa \in [k]^2$, with $|B_\kappa| = T$. We keep $\mathcal{C}$ unchanged and modify $\mathcal{T}$ to a function $\mathcal{T}'$ that maps from $\mathcal{C}$ into the power-set of $[n']^2$: For $\kappa \in [k]^2$ and $v \in [n]$, we simply add $T - T_{\kappa,v}$ arbitrary distinct dummy elements from $\{(f, v) \mid f \in B_\kappa\}$ to $\mathcal{T}(\kappa)$ in order to obtain $\mathcal{T}'(\kappa)$.

This ensures the vertical balance property defined in the statement of the lemma, and we observe that $\mathcal{T}'$ has the same grid tilings as $\mathcal{T}$: Every grid tiling of $\mathcal{T}$ is also one of $\mathcal{T}'$. Furthermore, dummy elements cannot be chosen in any grid tiling of $\mathcal{T}'$ since, for all κ and κ', the dummy elements in $\mathcal{T}'(\kappa)$ and $\mathcal{T}'(\kappa')$ have disjoint first coordinates, which are also distinct from $[n]$. Thus, in particular, any assignment using dummy elements cannot satisfy the first condition of a grid tiling required in Definition 2.4. □

3 Holants, matchgates, linear combinations of signatures

In the following, we give a introduction to what we call the *Holant framework*, a toolbox introduced by [53, 8, 9]. Some of this material is abridged from [15]. We use Holant problems as an intermediate step for reducing problems, such as counting grid tilings, to the permanent.

3.1 Signature graphs and Holants

The input to a Holant problem is a so-called signature graph, that is, a graph with certain functions associated with its vertices.

Definition 3.1. A *signature graph* is an edge-weighted graph Ω which may feature parallel edges, and which has a *vertex function* $f_v : \{0,1\}^{I(v)} \to \mathbb{C}$ associated with each $v \in V(\Omega)$. We also call f_v the *signature* of v. If v has degree d and an edge-ordering $I(v) = \{e_1, \ldots, e_d\}$ is specified, we also consider $f_v : \{0,1\}^d \to \mathbb{C}$.

The *Holant* of Ω is a particular sum over edge assignments $x \in \{0,1\}^{E(\Omega)}$. For $x \in \{0,1\}^{E(\Omega)}$, we say that $e \in E(\Omega)$ is *active in* x if $x(e) = 1$ holds, and we tacitly identify x with the set of active edges in x. Given a subset $S \subseteq E(\Omega)$, we write $x|_S$ for the restriction of x to S, which is the unique assignment in $\{0,1\}^S$ that agrees with x on S.

Definition 3.2 (adapted from [53])**.** Let Ω be a signature graph with edge weights $w : E(\Omega) \to \mathbb{C}$ and a vertex function $f_v : \{0,1\}^{I(v)} \to \mathbb{C}$ for each $v \in V(\Omega)$. For $x \in \{0,1\}^{E(\Omega)}$, we define

$$\mathrm{val}_\Omega(x) := \prod_{v \in V(\Omega)} f_v(x|_{I(v)}), \tag{1}$$

$$w_\Omega(x) := \prod_{e \in x} w(e). \tag{2}$$

We say that x *satisfies* Ω if $\mathrm{val}_\Omega(x) \neq 0$ holds. Furthermore, we define

$$\mathrm{Holant}(\Omega) := \sum_{x \in \{0,1\}^{E(\Omega)}} w_\Omega(x) \cdot \mathrm{val}_\Omega(x). \tag{3}$$

A particularly useful type of vertex functions is that of *Boolean functions*, whose ranges are restricted to $\{0,1\}$ rather than $\mathbb{C}$. If all signatures appearing in a signature graph Ω' are Boolean, then $\mathrm{Holant}(\Omega')$ simply sums over those assignments $x \in \{0,1\}^{E(\Omega')}$ that satisfy all constraints imposed by the vertex functions, and each x is weighted by $w_{\Omega'}(x)$. As an example, we use Boolean functions to reformulate PerfMatch as a Holant problem.

Example 3.3. *Given an edge-weighted graph G, let $f_v : \{0,1\}^{I(v)} \to \{0,1\}$ for $v \in V(G)$ be the vertex function defined by*

$$f_v(x) = \begin{cases} 1 & \textit{if } \mathrm{hw}(x) = 1, \\ 0 & \textit{otherwise.} \end{cases} \tag{4}$$

Let Ω denote the signature graph obtained from G by associating f_v with v, for all $v \in V(G)$. Then $\mathrm{Holant}(\Omega)$ *ranges over those assignments $x \in \{0,1\}^{E(\Omega)}$ in which each vertex is incident with exactly one active edge. Each such x is weighted by $w_\Omega(x) = \prod_{e \in x} w(e)$. This is precisely the expression of* $\mathrm{PerfMatch}(G)$*.*

3.2 Gates and matchgates

In some occasions, we can simulate signatures f appearing in a signature graph Ω by gadgets, i.e., signature graphs on "basic" signatures that realize f. We call such gadgets *gates*, similar to the $\mathcal{F}$-gates in [9], and we will be particularly interested in *matchgates*. These are gates Γ that feature, at each vertex $v \in V(\Gamma)$, the perfect matching signature from Example 3.3 that maps $x \in \{0,1\}^{I(v)}$ to

$$\mathtt{HW}_{=1}(x) := [\mathrm{hw}(x) = 1].$$

The formal definition of gates and matchgates follows.

Definition 3.4. A *gate* is a signature graph Γ containing a set $D \subseteq E(\Gamma)$ of *dangling edges*, all of which have edge-weight 1. A dangling edge is an "edge" that is incident with only one vertex. We consider the dangling edges of Γ to be labeled as $1, \ldots, |D|$.

Given a signature graph Ω, a vertex $v \in V(\Omega)$ of degree $|D|$, and an ordering of $I(v)$ as $I(v) = \{e_1, \ldots, e_{|D|}\}$, we can *insert* Γ at v by deleting v, placing a copy of Γ into G, and identifying e_i with the i-labeled dangling edge of Γ, for all i.

For disjoint sets A, B, and for $x \in \{0,1\}^A$ and $y \in \{0,1\}^B$, write $xy \in \{0,1\}^{A \cup B}$ for the assignment that agrees with x on A, and with y on B. We say that xy *extends* x. The *signature of* Γ is the function $\mathrm{Sig}(\Gamma) : \{0,1\}^D \to \mathbb{C}$ that maps $x \in \{0,1\}^D$ to

$$\mathrm{Sig}(\Gamma, x) = \sum_{y \in \{0,1\}^{E(\Gamma) \setminus D}} w_\Gamma(xy) \cdot \mathrm{val}_\Gamma(xy). \tag{5}$$

We also say that Γ *realizes* $\mathrm{Sig}(\Gamma)$. If all $v \in V(\Gamma)$ feature the function $\mathtt{HW}_{=1}$ defined above, then Γ is a *matchgate*. Finally, we call Γ planar if it can be drawn in the plane with all dangling edges on the outer face, such that they appear in the order $1, \ldots, |D|$ in a clockwise traversal of this face.

By the following lemma, if Γ realizes a signature f, and v is a vertex with signature f in a signature graph Ω, then we can insert Γ at v in a way that preserves Holants. In other words, we can treat Γ as if it were a single vertex of signature $\mathrm{Sig}(\Gamma)$. This will be used to reduce $\mathrm{Holant}(\Omega)$ to PerfMatch if all signatures in Ω can be realized by matchgates. For a proof, see Chapter 2 of [15].

Lemma 3.5. *Let Ω be a signature graph, let $v \in V(\Omega)$ be arbitrary, and let f_v denote the vertex function of v in Ω. Furthermore, let Γ be a (match-)gate with $\mathrm{Sig}(\Gamma) = f_v$, and let Ω' be obtained from Ω by inserting Γ at v. Then we have*

$$\mathrm{Holant}(\Omega) = \mathrm{Holant}(\Omega').$$

If Ω and Γ are planar and Ω is given together with a plane embedding, then the following holds: If we order $I(v)$ according to its clockwise ordering in the embedding and insert Γ under this order, then Ω' is planar.

In the remainder of this subsection, we consider specific matchgates that will be relevant later. To simplify our presentation, we abbreviate the following 4-bitstrings. Each corresponds to a specific assignment to the edges incident with a vertex of degree 4.

$$:= 0000, \qquad := 0101, \qquad := 1010, \qquad := 1111,$$
$$:= 1000, \qquad := 0010, \qquad := 1101, \qquad := 0111.$$

In Figure 1, we define a signature `PASS` of arity 4 and two signatures `PRE` and `ACT` of arity 6. Note that `PASS` essentially acts as a "crossing" signature: It enforces equality on its western and eastern dangling edges (numbered 4 and 2), as well as on its northern and southern dangling edges (numbered 1 and 3). However, if all dangling edges are active, then the output of `PASS` is -1 rather than 1. This flipped sign allows `PASS` to admit a planar matchgate $\Gamma_{\texttt{PASS}}$, shown in Figure 1. We verified that $\mathrm{Sig}(\Gamma_{\texttt{PASS}}) = \texttt{PASS}$ holds by means of a computer program: For all $x \in \{0,1\}^4$, we showed mechanically that $\mathrm{Sig}(\Gamma_{\texttt{PASS}}, x) = \texttt{PASS}(x)$ holds. Note that this verification can also be carried out by hand. For more details, consider Appendix C of [15]. It should also be noted that planar matchgates for `PASS` were already studied in [53, 6].

Next, we consider the signatures `PRE` and `ACT`, each of arity 6. We consider their last two inputs (the dangling edges with numbers 5 and 6) as "switches", which will later be connected to apices. It is crucial to observe that

$$\texttt{PRE}(x00) = \texttt{ACT}(x00) = \texttt{PASS}(x) \quad \forall x \in \{0,1\}^4.$$

That is, if the two switch edges are not active, then `PRE` and `ACT` behave exactly like `PASS` on their non-switch inputs. If both switches are active, then some differences occur, namely, the restriction to non-switch edges must be in state or for `PRE` or `ACT` to yield a nonzero value. Furthermore, if only one of the two switches is active, then `ACT` yields value zero, while `PRE` still allows such assignments (such as 01). We verified with a computer program that $\texttt{PRE} = \mathrm{Sig}(\Gamma_{\texttt{PRE}})$ holds for the matchgate $\Gamma_{\texttt{PRE}}$ from Figure 1. In the following, we prove manually that $\texttt{ACT} = \mathrm{Sig}(\Gamma_{\texttt{ACT}})$ holds.

Lemma 3.6. *We have $\texttt{ACT} = \mathrm{Sig}(\Gamma_{\texttt{ACT}})$ with the matchgate $\Gamma_{\texttt{ACT}}$ from Figure 1.*

Proof. Note that $\Gamma_{\texttt{ACT}}$ has a green vertex of signature `PRE`, and some additional part (a ring of `PASS` signatures, and an edge of weight $\frac{1}{2}$) which we call the *even filter*. Observe also that, for all $x \in \{0,1\}^4$ and $y \in \{0,1\}^2$, we have the identity

$$\texttt{PRE}(xy) = \begin{cases} \texttt{ACT}(xy) & \text{if } \mathrm{hw}(x) \text{ even,} \\ \text{arbitrary} & \text{otherwise.} \end{cases} \tag{6}$$

The even filter now ensures the following, for all $x \in \{0,1\}^4$ and $y \in \{0,1\}^2$:

- If $\mathrm{hw}(x)$ is not even, then $\mathrm{Sig}(\Gamma_{\texttt{ACT}}, xy) = 0$, regardless of the value of `PRE` on xy.
- If $\mathrm{hw}(x)$ is even, then $\mathrm{Sig}(\Gamma_{\texttt{ACT}}, xy) = \texttt{PRE}(xy)$. By (6), this implies $\mathrm{Sig}(\Gamma_{\texttt{ACT}}, xy) = \texttt{ACT}(xy)$.

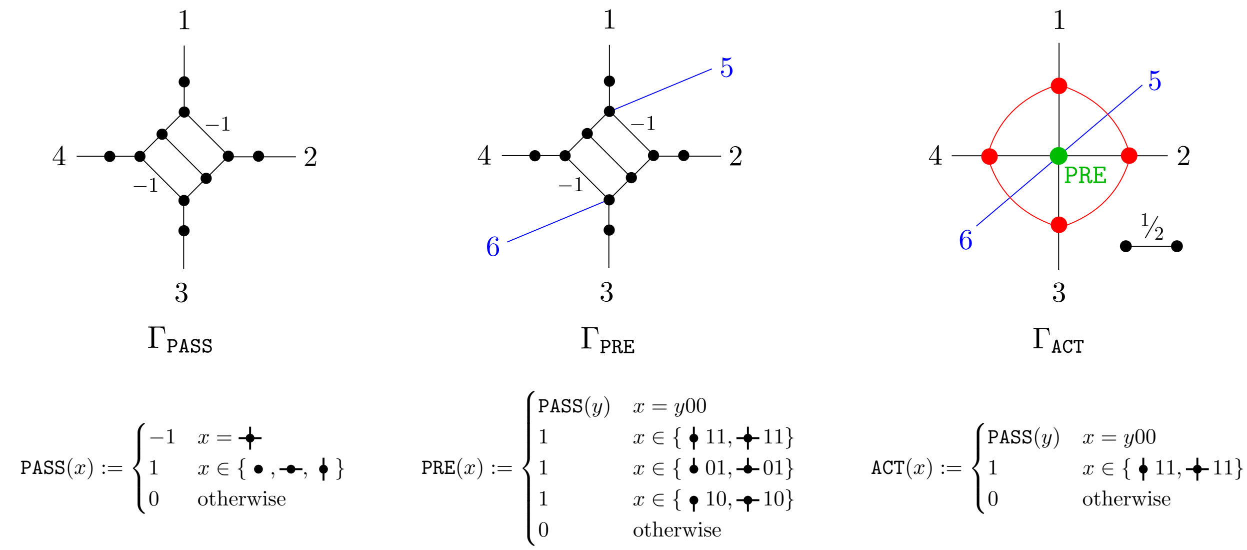

Figure 1: The matchgates $\Gamma_{\texttt{PASS}}$, $\Gamma_{\texttt{PRE}}$ and $\Gamma_{\texttt{ACT}}$ and the signatures $\texttt{PASS}$, $\texttt{PRE}$ and $\texttt{ACT}$. Note that $\Gamma_{\texttt{PASS}}$ has four dangling edges, numbered 1 to 4, whereas $\Gamma_{\texttt{PRE}}$ and $\Gamma_{\texttt{ACT}}$ each have six dangling edges, numbered 1 to 6. The signature $\texttt{PASS}$ is defined on assignments $x \in \{0,1\}^4$, while $\texttt{PRE}$ and $\texttt{ACT}$ are defined on assignments $x \in \{0,1\}^6$. These strings correspond canonically to assignments at the dangling edges of $\Gamma_{\texttt{PASS}}$, $\Gamma_{\texttt{PRE}}$ and $\Gamma_{\texttt{ACT}}$. All black vertices are assigned $\texttt{HW}_{=1}$. In the gate $\Gamma_{\texttt{ACT}}$, all red vertices are assigned $\texttt{PASS}$, and the green middle vertex is assigned $\texttt{PRE}$. Note that we can also view $\Gamma_{\texttt{ACT}}$ as a matchgate by realizing its signatures with the matchgates $\Gamma_{\texttt{PASS}}$ and $\Gamma_{\texttt{ACT}}$. All matchgates are planar after removal of the dangling edges 5 and 6, which will later connect to apex vertices.

Since $\texttt{ACT}(xy) \neq 0$ implies $x \in \{ , , , \}$, which in turn implies that $\mathrm{hw}(x)$ is even, this will prove the lemma. To compute $\mathrm{Sig}(\Gamma_{\texttt{ACT}}, xy)$ for $x \in \{0,1\}^4$ and $y \in \{0,1\}^2$, we consider the satisfying assignments w to $E(\Gamma_{\texttt{ACT}})$ that extend xy. The dummy edge of weight $1/2$ is present in any assignment w and contributes a factor $1/2$ to $\mathrm{val}(w)$. (In this proof, we write $\mathrm{val}(w)$ instead of $\mathrm{val}_{\Gamma_{\texttt{ACT}}}(w)$ to avoid double indexing.) At each red vertex, the signature $\texttt{PASS}$ ensures that opposing edges have the same assignment under w. This fixes the value of all black edges and ensures that $\mathrm{val}(w)$ contains the factor $\texttt{PRE}(xy)$, contributed from the green vertex with signature $\texttt{PRE}$.

It remains to assign values to the red edges: Due to the signature $\texttt{PASS}$ at red vertices, this is possible with at most two satisfying assignments $w_1, w_2 \in \{0,1\}^{E(\Gamma_{\texttt{ACT}})}$:

w_1 : All red edges are active. Then every red vertex in state yields a factor $\texttt{PASS}( ) = -1$, while all other red vertices are in one of the states or and yield value 1. The number of red vertices in state is $\mathrm{hw}(x)$, so the value of $\Gamma_{\texttt{ACT}}$ on w_1 is

$$\mathrm{val}(w_1) = \frac{1}{2} \cdot (-1)^{\mathrm{hw}(x)} \cdot \texttt{PRE}(xy).$$

w_2 : No red edges are active. Then every red vertex is in one of the states or and hence yields value 1. Thus, the value of $\Gamma_{\texttt{ACT}}$ on w_2 is

$$\mathrm{val}(w_2) = \frac{1}{2} \cdot \texttt{PRE}(xy).$$

It follows that for all $x \in \{0,1\}^4$ and $y \in \{0,1\}$, we have

$$\begin{aligned}
\mathrm{Sig}(\Gamma_{\mathtt{ACT}}, xy) &= \mathrm{val}(w_1) + \mathrm{val}(w_2) \\
&= \frac{1}{2} \cdot \Big((-1)^{\mathrm{hw}(x)} \cdot \mathtt{PRE}(xy) + \mathtt{PRE}(xy) \Big) \\
&= \begin{cases} \mathtt{PRE}(xy) & \text{if } \mathrm{hw}(x) \text{ even}, \\ 0 & \text{otherwise.} \end{cases} \\
&= \mathtt{ACT}(xy)
\end{aligned}$$

This proves the lemma. □

3.3 Linear combinations of matchgate signatures

We introduce our main tool for the later sections, a technique that allows us to simulate signatures by linear combinations of other signatures, in particular, of matchgate signatures.

Definition 3.7. Let $f = c_1 \cdot f_1 + \ldots + c_t \cdot f_t$ be a signature, where $c_1, \ldots, c_t \in \mathbb{C}$ are coefficients and $f_1, \ldots, f_t$ are signatures, and the linear combination is point-wise. Then we say that f is t-combined from constituents $f_1, \ldots, f_t$.

We apply such linear combinations as follows: Assume we are given a signature graph that features k occurrences of some interesting signature f which cannot be realized by matchgates. If we can express f as a linear combination of t constituents that do admit matchgates, then the following lemma allows us to compute Holant(Ω) from the Holants of t^k derived signature graphs whose signatures all admit matchgates.

Lemma 3.8. *Let Ω be a signature graph, let $k, t \in \mathbb{N}$ and let $w_1, \ldots, w_k$ be distinct vertices of Ω such that the following holds: For all $\kappa \in [k]$, the signature f_κ at w_κ admits coefficients $c_{\kappa,1}, \ldots, c_{\kappa,t} \in \mathbb{C}$ and signatures $g_{\kappa,1}, \ldots, g_{\kappa,t}$ such that $f_\kappa = \sum_{i=1}^{t} c_{\kappa,i} \cdot g_{\kappa,i}$. Given a tuple $\theta \in [t]^k$, let Ω_θ be defined by replacing, for each $\kappa \in [k]$, the vertex function f_κ at w_κ with $g_{\kappa,\theta(\kappa)}$. Then we have*

$$\mathrm{Holant}(\Omega) = \sum_{\theta \in [t]^k} \left(\prod_{\kappa=1}^{k} c_{\kappa,\theta(\kappa)} \right) \cdot \mathrm{Holant}(\Omega_\theta). \tag{7}$$

Proof. Choose any fixed single $\kappa \in [k]$. For $i \in [t]$, let Ω_i denote the signature graph obtained from Ω by replacing f_κ with $g_{\kappa,i}$. By elementary manipulations, we have

$$\begin{aligned}
\mathrm{Holant}(\Omega) &= \sum_{x \in \{0,1\}^{E(\Omega)}} f_\kappa(x) \cdot \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\
&= \sum_{x \in \{0,1\}^{E(\Omega)}} \left(\sum_{i=1}^{t} c_{\kappa,i} \cdot g_{\kappa,i}(x) \right) \cdot \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\
&= \sum_{i=1}^{t} c_{\kappa,i} \cdot \sum_{x \in \{0,1\}^{E(\Omega)}} g_{\kappa,i}(x) \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\
&= \sum_{i=1}^{t} c_{\kappa,i} \cdot \mathrm{Holant}(\Omega_i).
\end{aligned}$$

Then apply this identity inductively for $\kappa = 1, \ldots, k$. Each step reduces the number of combined signatures by one, and elementary algebraic manipulations imply (7). □

When using Lemma 3.8 for positive results, as in Section 4, then the right-hand side of (7) is "easy", in the sense that the values $\mathrm{Holant}(\Omega_\theta)$ for all θ can be obtained efficiently, e.g., by reduction to planar PerfMatch. In the same way, Lemma 3.8 also allows us to prove hardness results under Turing reductions, as we do in Sections 5 and 6: In this case, the left-hand side is "hard" and could be computed from oracle access to the values $\mathrm{Holant}(\Omega_\theta)$ for all θ.

4 PerfMatch on bounded-genus graphs

In this section, we present a first application of the framework of combined signatures: We show that, for graphs of genus k, the quantity $\mathrm{PerfMatch}(G)$ can be expressed as a linear combination of 4^k values $\mathrm{PerfMatch}(G_i)$, where G_i is a planar graph for all $i \in [4^k]$. The linear combinations resemble those used in [29, 49, 44], but unlike these papers, we can state our linear combinations without any necessity for Pfaffian orientations. That is, we obtain a parameterized reduction with black-box access to counting perfect matchings in planar graphs.

4.1 The algorithm

Following [49], we assume that the graph G in question is given to us together with a plane model: All vertices of G are drawn in a polygon P with $2k$ sides. If there is a set of d_i parallel edges $x_i = x_{i1}x_{i2}\cdots x_{id_i}$ leaving P from one side and going into P through another side, we denote the two sides by a_i and a_i^{-1} respectively. Since the edges are parallel, when we walk along the sides of P counterclockwise, we meet the exits of edges in the order $x_{i1}x_{i2}\cdots x_{id_i}$ on side a_i, then the entrances of edges in the order $x_{id_i}x_{i(d_i-1)}\cdots x_{i1}$ on side a_i^{-1}. If G can be embedded on an orientable compact boundaryless surface S of genus k, then it can be drawn such that there are no edges crossing inside P, and the sides of P are

$$a_1a_2a_1^{-1}a_2^{-1}a_3a_4a_3^{-1}a_4^{-1}\cdots a_{2k-1}a_{2k}a_{2k-1}^{-1}a_{2k}^{-1}.$$

The side pair a_i, a_i^{-1} represents boundaries to be glued together. When G is drawn on the surface S, the edge bunches x_1 and x_2 overpass each other without any edges crossing; see the left picture of Figure 2 for such a situation, which we call a *grid cap*.

We use linear combinations of matchgates to simulate the grid cap by a planar graph. Write x_i^{-1} to denote $x_{id_i}x_{i(d_i-1)}\cdots x_{i1}$. Then the grid cap realizes a function that is defined on assignments (x_1,x_2,y_1,y_2) to its dangling edges as follows:

$$O(x_1,x_2,y_1,y_2) = [y_1 = x_1^{-1}]\cdot[y_2 = x_2^{-1}].$$

The straightforward idea is to place a PASS matchgate at each crossing of overpassing edges, as shown in the middle of Figure 2. Let us denote by $C(x_1,x_2,y_1,y_2)$ the signature of the resulting gate. In any satisfying assignment (x_1,x_2,y_1,y_2) to its dangling edges, there are $\mathrm{hw}(x_1)\cdot\mathrm{hw}(x_2)$ instances of PASS in state , each of which gives a factor -1, while all other instances of PASS (in states , ,) give a factor 1, so

$$C(x_1,x_2,y_1,y_2) = (-1)^{\mathtt{ODD}(x_1)\cdot\mathtt{ODD}(x_2)}\cdot[y_1 = x_1^{-1}]\cdot[y_2 = x_2^{-1}].$$

We can therefore conclude that O can be expressed as a linear combination of signatures of type C, each of which is the signature of a planar matchgate.

Lemma 4.1. *Every grid cap gate is a linear combination of* 4 *matchgates, given by*

$$O(x_1,x_2,y_1,y_2) = \frac{1}{2}(1 + (-1)^{\mathtt{ODD}(x_1)} + (-1)^{\mathtt{ODD}(x_2)} + (-1)^{\mathtt{ODD}(x_1)+\mathtt{ODD}(x_2)+1})\cdot C(x_1,x_2,y_1,y_2).$$

Proof. Observe first that

$$O(x_1,x_2,y_1,y_2) = \frac{1}{2}(1 + (-1)^{\mathtt{ODD}(x_1)} + (-1)^{\mathtt{ODD}(x_2)} + (-1)^{\mathtt{ODD}(x_1)+\mathtt{ODD}(x_2)+1})\cdot(-1)^{\mathtt{ODD}(x_1)\cdot\mathtt{ODD}(x_2)}.$$

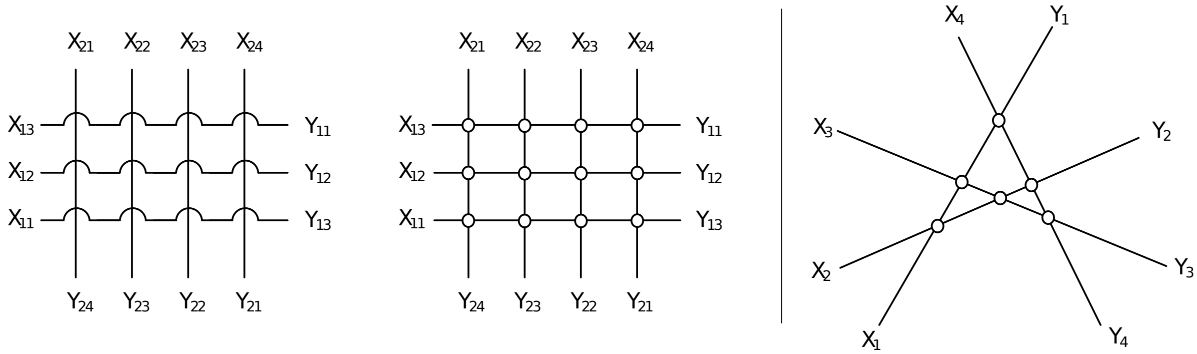

Figure 2: The first two subfigures show a grid cap and the matchgate realizing one of the constituents used to realize the grid cap. The third subfigure shows the matchgate used to simulate a cross cap. In these matchgates, all vertices are assigned the signature PASS.

From this, we can conclude that

$$\begin{aligned} O(x_1,x_2,y_1,y_2) &= \frac{1}{2}C(x_1,x_2,y_1,y_2) + \frac{1}{2}(-1)^{\texttt{ODD}(x_1)}C(x_1,x_2,y_1,y_2) + \\ &+ \frac{1}{2}(-1)^{\texttt{ODD}(x_2)}C(x_1,x_2,y_1,y_2) - \frac{1}{2}(-1)^{\texttt{ODD}(x_1)}(-1)^{\texttt{ODD}(x_2)}C(x_1,x_2,y_1,y_2). \end{aligned}$$

The extra factor $(-1)^{\texttt{ODD}(x_1)}$ can be realized by giving weight -1 instead of 1 to each edge x_{1i} in the matchgate C. Hence, all the four functions can be realized by some matchgates similar to C after introduction of additional -1 weights at some edges. □

We now consider non-orientable surfaces and their plane models: If G can be embedded on a non-orientable surface S, which is the connected sum of a surface of orientable genus k with either a projective plane or a Klein bottle, then it can be drawn without crossings inside P, such that the sides of P are

$$a_1a_2a_1^{-1}a_2^{-1}a_3a_4a_3^{-1}a_4^{-1}\cdots a_{2k-1}a_{2k}a_{2k-1}^{-1}a_{2k}^{-1}a_{2k+1}a_{2k+2}, \text{ and}$$

$$a_1a_2a_1^{-1}a_2^{-1}a_3a_4a_3^{-1}a_4^{-1}\cdots a_{2k-1}a_{2k}a_{2k-1}^{-1}a_{2k}^{-1}a_{2k+1}a_{2k+2}a_{2k+3}a_{2k+4},$$

respectively. Here, the side pair a_ia_i means that, when a bunch of edges $x_i = x_{i1}x_{i2}\cdots x_{id_i}$ leaves the interior of P through the first side a_i and then enters back into P through the second side a_i, then we meet the exits and entrances in the order x_ix_i. Such a bunch of edges is called a *cross cap*, and it realizes a function

$$O(x,y) = [y = x].$$

If we draw it on the plane and replace each crossing by a PASS matchgate, as shown in the right part of Figure 2, we get a matchgate realizing

$$C(x,y) = (-1)^{\binom{\mathrm{hw}(x)}{2}} \cdot [y = x].$$

From this, we obtain a linear combination for cross cap gates from planar matchgates:

Lemma 4.2. *Every cross cap gate is a linear combination of* 2 *matchgates, given by*

$$O(x,y) = \frac{1-i}{2} \cdot i^{\mathrm{hw}(x)} \cdot C(x,y) + \frac{1+i}{2} \cdot (-i)^{\mathrm{hw}(x)} \cdot C(x,y).$$

Proof. The sequence $(-1)^{\binom{\mathrm{hw}(x)}{2}}$ indexed by $\mathrm{hw}(x)$ is

$$1, 1, -1, -1, 1, 1, -1, -1, \ldots$$

It must be a linear combination of 4 sequences $w^{\mathrm{hw}(x)}$, for $w \in \{1, i, -1, -i\}$, all of which have the same period 4, since the length 4 initial segments of the 4 sequences form a full rank Vandermonde matrix. In fact, it can be expressed as a linear combination of two such sequences, as we can observe that

$$(-1)^{\binom{\mathrm{hw}(x)}{2}} = \frac{1-i}{2} i^{\mathrm{hw}(x)} + \frac{1+i}{2} (-i)^{\mathrm{hw}(x)}.$$

The extra factor $i^{\mathrm{hw}(x)}$ can be realized by giving weight i instead of 1 to each input edge in C. □

Using the fact that G is embedded as a plane model, and using the combined signatures for grid caps and cross caps from the last two lemmas, we then obtain the following known theorem.

Theorem 4.3. *[49] Let G be a graph that is embedded on a surface. Then* $\mathrm{PerfMatch}(G)$ *is a summation of* PerfMatch *of* 2^{2k}, 2^{2k+1} *or* 2^{2k+2} *planar graphs, respectively, if the surface is the connected sum of an orientable surface of genus k with the plane, the projective plane, or the Klein bottle, respectively.*

Proof. By Lemma 4.1 and 4.2, use Lemma 3.8 on the k grid caps and 0, 1 or 2 cross caps. □

4.2 Additional remarks

For a matrix A, let $A^{\otimes k}$ denote the matrix obtained from the k-fold Kronecker product $A \otimes \ldots \otimes A$. The essence of Lemma 4.1 is that we can use the four matchgates to realize all four columns of the basis

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes 2},$$

so that we can then obtain any other function by linear combinations. The same observation also holds for a larger base

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes m}.$$

We give an example: In a cross cap of m edges, we may replace each edge by a bunch of parallel edges, and call the result a *grated cross cap*. All the $\binom{m}{2}$ latent crossings of the cross cap become grid caps in the grated cross cap.

Fact 4.4. *Every grated cross cap gate over m bunches of edges, as defined above, can be expressed as a linear combination of* 2^m *planar matchgates.*

In fact, these 2^m basis matchgates are powerful enough to express (as a linear combination) any function that depends only upon the parities $p_1, \ldots, p_m$ of active edges in the m edge bunches. However, among these functions, we currently only know one interesting function, i.e., the grid cap. Even the grated cross cap seems too artificial to be related with a natural tractability result. A similar generalization applies to Lemma 4.2, where the functions to be expressed may also depend upon residuals of the numbers of active edges in the m edge bunches, in this case however modulo 4 rather than 2.

5 The permanent on k-apex graphs

In this section, we prove Theorem 1.1 by an application of our framework of combined signatures. We use #GridTiling as a reduction source, and from a high level, our approach could be compared to, say, the reduction in [39] for planar multiway cut. Given an instance $\mathcal{A}$ to #GridTiling, we proceed as follows:

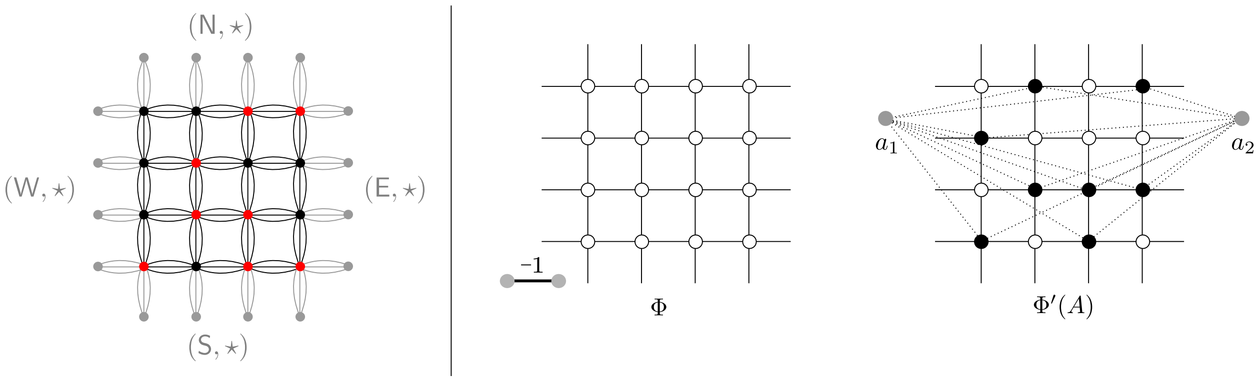

Figure 3: The left part of the figure shows the signature graph $G(\mathcal{A})$. Border vertices c_κ for $\kappa \in \{\mathsf{N},\mathsf{W},\mathsf{S},\mathsf{E}\}\times[k]$ and their incident edges are colored gray. Cell vertices c_κ for $\kappa \in \mathcal{C}$ are colored red, while vertices c_κ for $\kappa \in [k]^2 \setminus \mathcal{C}$ are colored black. Horizontally or vertically adjacent vertices are connected by an edge bundle of n parallel edges. The right part of the figure shows the gates Φ and $\Phi'(A)$. Each white vertex is assigned `PASS`, each black vertex is assigned `ACT`, and each gray vertex is assigned $\mathtt{HW}_{=1}$. Edges from apices in Φ' are drawn dashed. Note that, due to the balance property of $\mathcal{T}$, we may assume that every column has the same number T of occurrences of `ACT`.

1. We express the solution to the instance as $\mathrm{Holant}(G)$ for a signature graph G in Section 5.1.
2. We realize the signatures of G in Section 5.2. At this point however, we require combined signatures, and this is where we depart from the usual reductions from GridTiling.

Large parts of this section will be reused in Section 6 with an added layer of technicalities.

5.1 Global construction

In the following, let $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ be a fixed instance to #GridTiling, as specified in Definition 2.4. By applying vertical balance as in Lemma 2.10, we may assume the existence of some number $T \leq n$ such that for all $\kappa \in \mathcal{C}$ and all $v \in [n]$, there are exactly T elements of type $(\star, v)$ in $\mathcal{T}(\kappa)$. This will become relevant in Section 5.2.

First, we reformulate $\mathcal{A}$ as the Holant of a signature graph $G = G(\mathcal{A})$. This graph G consists of a $k \times k$ square grid of *cells,* and $4k$ additional *border vertices* adjacent to the borders of the grid, as seen in the left part of Figure 3. Note that G is planar. We denote its vertices by c_κ for $\kappa \in \Xi$, where

$$\Xi := [k]^2 \cup \{\mathsf{N},\mathsf{W},\mathsf{S},\mathsf{E}\} \times [k].$$

For $i \in [k]$, we declare (N, i) to be vertically adjacent to $(1, i)$, and (S, i) to (k, i). Likewise, we declare (W, i) to be horizontally adjacent to $(i, 1)$, and (E, i) to (i, k). We refer to the neighbors of any index $\kappa \in \Xi$ or vertex $c_\kappa \in V(G)$ using cardinal directions in the obvious way, e.g., we may speak of the northern neighbor of a vertex. Between any pair of vertices c_κ and $c_{\kappa'}$ with adjacent indices κ and κ', we place a set $E_{\kappa,\kappa'}$ of n parallel edges, which we call an *edge bundle.*

We proceed to define the signatures of G. In the assignments $a \in \{0,1\}^{E(G)}$ we are interested in, each edge bundle features exactly one active edge, which is used to encode a number from $[n]$. At border vertices, we place the signature $\mathtt{HW}_{=1}$ to ensure this. The signatures of cells c_κ with $\kappa \in [k]^2$ are then defined so that each cell propagates the number $x_W \in [n]$ encoded by its western incident edge bundle to the east, and its northern number $x_N \in [n]$ to the south, while checking along the way whether $(x_W, x_N) \in \mathcal{T}(\kappa)$ holds.

Remark 5.1. We adhere to the following notational conventions in this section:

- For $v \in [n]$, we often identify the string $0^{v-1}10^{n-v} \in \{0,1\}^n$ with the number v when it is clear from the context which of these two objects we currently refer to.

- For $\kappa \in [k]^2$, the $4n$ incident edges of each vertex c_κ are ordered such that all northern edges appear first, in a block of length n, followed by the n eastern, the n southern, and finally the n western edges.
- We implicitly consider strings $x \in \{0,1\}^{4n}$ to be decomposed into $x = x_N x_E x_S x_W$ with four bistrings $x_N, x_E, x_S, x_W \in \{0,1\}^n$ corresponding to the four cardinal directions.

Using these conventions, we then define the following predicates for strings $x \in \{0,1\}^{4n}$:

$$\begin{aligned}\varphi_{one}(x) &\equiv \mathrm{hw}(x_N) = 1 \wedge \mathrm{hw}(x_W) = 1,\\ \varphi_{prop}(x) &\equiv x_N = x_S \wedge x_W = x_E.\end{aligned}$$

If a function f satisfies $\varphi_{prop}(x)$ for each $x \in \mathrm{supp}(f)$, then we call f *propagating*. For each $\kappa \in [k]^2$, we place a specific propagating signature f_κ at the vertex c_κ in order to complete G to a signature graph whose satisfying assignments correspond bijectively to the grid tilings of $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$.

Definition 5.2. Let let $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ an instance to the grid tiling problem, as described above. For all $\kappa \in [k]^2 \setminus \mathcal{C}$, we define the vertex function $f_\kappa : \{0,1\}^{4n} \to \{0,1\}$ of c_κ such that, for all $x \in \{0,1\}^{4n}$ satisfying the predicate $\varphi_{one}(x)$, we have

$$f_\kappa(x) := [\varphi_{prop}(x)].$$

Note that no requirement is imposed upon $f_\kappa(x)$ on those $x \in \{0,1\}^{4n}$ that fail to satisfy $\varphi_{one}(x)$. For all remaining κ, namely all $\kappa \in \mathcal{C}$, we define the vertex function g_κ of c_κ on such $x \in \{0,1\}^{4n}$ by declaring

$$g_\kappa(x) := [\varphi_{prop}(x) \wedge (x_W, x_N) \in \mathcal{T}(\kappa)]$$

This finishes the definition of the signature graph $G = G(\mathcal{A})$. In the following, we verify by a simple argument that G indeed encodes $\mathcal{A}$ properly.

Lemma 5.3. *The grid tilings of $\mathcal{A}$ correspond bijectively to the satisfying assignments $x \in \{0,1\}^{E(G)}$ of G, and each satisfying assignment x additionally has $\mathrm{val}_G(x) = 1$.*

Proof. Every grid tiling $a : [k]^2 \to [n]^2$ can be transformed into an assignment $x(a) \in \{0,1\}^{E(G)}$ as follows: For each $\kappa \in [k]^2$, with $a(\kappa) = (u, v)$, declare the u-th edge in the western edge bundle of c_κ and the v-th edge in the northern edge bundle of c_κ to be active. At vertices $c_{(k,\star)}$, copy the assignment from northern edges to southern edges, and at $c_{(\star,k)}$, copy the assignment from western edges to eastern edges. Declare all other edges to be inactive. It follows from the definition of f_κ at $\kappa \in \mathcal{C}$ and g_κ at $\kappa \in [k]^2 \setminus \mathcal{C}$ that $\mathrm{val}_G(x(a)) = 1$ holds.

For the converse direction, we show that every satisfying assignment $x \in \{0,1\}^{E(G)}$ can be written as $x = x(a)$ for some grid tiling a, where $x(a)$ is defined as in the previous paragraph. Note that this also implies $\mathrm{val}_G(x) = 1$. By the signature $\mathtt{HW}_{=1}$, every border vertex is incident with exactly one active edge in x. Hence, the restriction of x to $I(c_{1,1})$ satisfies φ_{one}; call this restricted assignment y.

- If $(1,1) \in [k]^2 \setminus \mathcal{C}$, then the vertex function of $c_{1,1}$ is $f_{1,1}$. Since $f_{1,1}(y) = 1$, and since $f_{1,1}$ is propagating on inputs satisfying φ_{one}, we also have $\varphi_{prop}(y)$.
- If $(1,1) \in \mathcal{C}$, then we additionally have $(y_W, y_N) \in \mathcal{T}(1,1)$ by definition of $g_{1,1}$.

By induction along rows and columns, we obtain, for every $\kappa \in [k]^2$, that the partial assignment y at $I(c_\kappa)$ satisfies $\varphi_{prop}(y)$ and $(y_W, y_N) \in \mathcal{T}(\kappa)$ if $\kappa \in \mathcal{C}$. Hence $x = x(a)$ holds for a unique grid tiling a. ☐

In the next subsection, we realize each signature f_κ for $\kappa \in \mathcal{C}$ as a planar matchgate, and each g_κ for $\kappa \in [k]^2 \setminus \mathcal{C}$ as a linear combination of two matchgate signatures that have maximum apex number 2. Note that the remaining signatures $\mathtt{HW}_{=1}$ occurring in G are planar. Since G itself is planar and features at most $\mathcal{O}(k)$ signatures g_κ, the graphs realizing G will feature at most $\mathcal{O}(k)$ apices, and we will use this to obtain the desired parameterized reduction and lower bound under #ETH.

5.2 Realizing cell signatures

It can be shown (under no additional assumptions) that some of the signatures g_κ for $\kappa \in [k]^2$ are non-planar. From a complexity viewpoint, if all such signatures were planar and we knew explicit planar matchgates, then we could reduce #GridTiling to planar PerfMatch, and thus show $\mathsf{FP} = \#\mathsf{P}$ by the FKT method. Rather than trying to use planar matchgates, we show that each signature g_κ can be realized as a specific *linear combination* of the signatures of one planar and one 2-apex matchgate. Note again that at least one non-planar constituent is necessary, as we could otherwise show $\mathsf{FPT} = \#\mathsf{W}[1]$.

In the remainder of this section, we consider $\kappa \in [k]^2$ to be fixed, we write $A = \mathcal{T}(\kappa)$ and we recall that $A \subseteq [n]^2$. The constituents for g_κ will be the signatures of two gates Φ and $\Phi'(A)$, which use as building blocks the signatures $\mathtt{PASS}$ and $\mathtt{ACT}$ from Section 3.

Definition 5.4. Let $n \in \mathbb{N}$ and let $A \subseteq [n]^2$. We define gates Φ and $\Phi' = \Phi'(A)$ with $4n$ dangling edges (that is, with n dangling edges for each cardinal direction) as follows. Consider also the right part of Figure 3.

- To obtain the gate Φ, arrange vertices b_τ for $\tau \in [n]^2$ in a $n \times n$ grid and assign the signature $\mathtt{PASS}$ to each such vertex. Add a single edge of weight -1 between two fresh vertices of signature $\mathtt{HW}_{=1}$.
- A similar construction yields the gate Φ': Starting from Φ, remove the extra edge of weight -1, add apex vertices a_1 and a_2 with signatures $\mathtt{HW}_{=1}$, and for all $\tau \in A$, do the following:
    1. Replace the signature $\mathtt{PASS}$ at b_τ with $\mathtt{ACT}$.
    2. Add the edges $a_1 b_\tau$ and $a_2 b_\tau$. Declare these to be the last two edges in the edge ordering of $I(v_\tau)$.

Recall that $\mathtt{PASS}$ is realized by the planar matchgate $\Gamma_{\mathtt{PASS}}$, so we can also view the gate Φ as a planar matchgate after realizing all signatures by matchgates. We will later switch between these views depending on the application. Note also that the 2-coloring of $\Gamma_{\mathtt{PASS}}$ can be extended to one of Φ. Likewise, $\mathtt{ACT}$ is realized by the matchgate $\Gamma_{\mathtt{ACT}}$, which is planar when ignoring its dangling edges 5 and 6. That is, after realizing each occurrence of $\mathtt{ACT}$ by $\Gamma_{\mathtt{ACT}}$, the resulting matchgate obtained from Φ' is planar after removal of a_1 and a_2.

Our goal for this subsection is to realize the signatures f_κ and g_κ from Definition 5.2. In the following, we prove that $f_\kappa = \mathrm{Sig}(\Phi)$ and that g_κ can be realized by a linear combination of $\mathrm{Sig}(\Phi)$ and $\mathrm{Sig}(\Phi')$. It will be crucial for our calculations to assume our instance $\mathcal{A}$ for GridTiling to be balanced: By Lemma 2.10, we assume there is some $T \in \mathbb{N}$ such that $|A \cap (\star, v)| = T$ for all $v \in [n]$. That is, in the right part of Figure 3, we may assume that every column of $\Phi'(A)$ features the same number T of vertices with signature $\mathtt{ACT}$.

Lemma 5.5. *Recall the definition of the predicates φ_{one} and φ_{prop} on the preceding page. Let $x \in \{0,1\}^{4n}$ be an assignment that satisfies the predicate φ_{one}. Then*

$$\mathrm{Sig}(\Phi, x) = \begin{cases} 0 & \text{if } \neg\varphi_{prop}(x), \\ 1 & \text{if } \varphi_{prop}(x). \end{cases} \tag{8}$$

$$\mathrm{Sig}(\Phi'(A), x) = \begin{cases} 0 & \text{if } \neg\varphi_{prop}(x) \\ \begin{cases} -T & \text{if } (x_W, x_N) \notin A \\ -T+2 & \text{if } (x_W, x_N) \in A \end{cases} & \text{if } \varphi_{prop}(x). \end{cases} \tag{9}$$

Note that $f_\kappa = \mathrm{Sig}(\Phi)$ for $\kappa \in [k]^2 \setminus \mathcal{C}$. For $\kappa \in \mathcal{C}$ and for $x \in \{0,1\}^{4n}$ satisfying φ_{one}, we have

$$g_\kappa(x) = \frac{T}{2} \cdot \mathrm{Sig}(\Phi, x) + \frac{1}{2} \cdot \mathrm{Sig}(\Phi'(\mathcal{T}(\kappa)), x). \tag{10}$$

In Section 5.3, we prove Lemma 5.5 by inspecting the possible satisfying assignments to Φ and Φ'. Before doing this, let us first show how Lemma 5.5 implies Theorem 1.2. We will require parts of this argument again in Section 6.

Proof of Theorem 1.2. Using Lemma 5.3, we know that $\mathrm{Holant}(G)$ counts precisely the grid tilings of $\mathcal{A}$. By Theorem 2.9, this problem is $\#\mathsf{W}[1]$-hard and cannot be solved in time $f(k) \cdot n^{o(k/\log k)}$, even on instances $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ with $|\mathcal{C}| = \mathcal{O}(k)$.

Using the linear combination (10) and Lemma 3.8 about the linear combinations of signatures, as well as Lemma 3.5 about inserting matchgates into signature graphs, we obtain

$$\mathrm{Holant}(G) = \frac{1}{2^{|\mathcal{C}|}} \sum_{\omega:\mathcal{C}\to[2]} T^{d(\omega)} \cdot \mathrm{PerfMatch}(H_\omega). \tag{11}$$

For $\omega : \mathcal{C} \to [2]$, the number $d(\omega)$ is the number of 1-entries in ω, and the graph H_ω is obtained as follows:

- For $\kappa \in [k]^2 \setminus \mathcal{C}$, insert the matchgate Φ at the cell vertex c_κ.
- For $\kappa \in \mathcal{C}$ with $\omega(\kappa) = 1$, insert the matchgate Φ at c_κ as well.
- For $\kappa \in \mathcal{C}$ with $\omega(\kappa) = 2$, insert the matchgate $\Phi'(\mathcal{T}(\kappa))$ at c_κ.

Since G is planar, and since Φ is planar and $\Phi'(\mathcal{T}(\kappa))$ for $\kappa \in \mathcal{C}$ has at most 2 apices, it follows that $\mathrm{apex}(H_\omega) \leq 2|\mathcal{C}|$ for all $\omega : \mathcal{C} \to [2]$, and this proves the required parameter bound. By 2-coloring the matchgates Φ and Φ', it can furthermore be verified that each graph H_ω is bipartite.

Additionally, by construction of the matchgates $\Gamma_{\mathtt{PASS}}$ and $\Gamma_{\mathtt{ACT}}$, every graph H_ω features only edge-weights from the set $\{-1, 1/2, 1\}$. Non-unit edge-weights in H_ω appear only at edges $uv \in E(H_\omega)$ not incident with apices. We can hence use standard weight simulation techniques to remove the edge-weights -1 and $1/2$, as in [51] or Chapter 1 of [15], while maintaining the apex number. We consequently obtain $\#\mathsf{W}[1]$-completeness of the permanent under the apex parameter and the claimed lower bound under $\#\mathsf{ETH}$. $\square$

Remark 5.6. The following might prove useful for later applications: By construction, the apices in the constructed graphs H_ω form an independent set, for any $\omega : [k]^2 \to [2]$, and each non-apex vertex in H_ω is incident with at most one apex. This last condition holds because the matchgate $\Gamma_{\mathtt{ACT}}$ has no vertex with two incident dangling edges.

5.3 Calculating the signatures of Φ and Φ'

In the remainder of this section, we provide the deferred proof of Lemma 5.5. To this end, we calculate the signatures of Φ and Φ' by analyzing, for any given assignment $x \in \{0,1\}^{4n}$ to their dangling edges, the possible satisfying assignments xy extending x.

5.3.1 Calculating the signature of Φ

Let $x \in \{0,1\}^{4n}$ be an assignment to the dangling edges of Φ that satisfies $\varphi_{one}(x)$, and let $xy \in \{0,1\}^{E(\Phi)}$ be a satisfying assignment to Φ that extends x. We show that, whenever $\varphi_{prop}(x)$ holds, then y is unique and xy has value 1, so $\mathrm{Sig}(\Phi, x) = \mathrm{val}_\Phi(xy) = 1$. Furthermore, we show that, if x does not satisfy the predicate φ_{prop}, then no such y exists, and hence $\mathrm{Sig}(\Phi, x) = 0$.

Recall from Remark 5.1 that we implicitly decompose the string x into x_N, x_E, x_S, x_W. Write $W \in [n]$ and $N \in [n]$ for the unique non-zero index in $x_W \in \{0,1\}^n$ and $x_N \in \{0,1\}^n$, respectively. These numbers are well-defined because x satisfies $\varphi_{one}(x)$ by assumption. Then all western and eastern edges of vertices in row $(W, \star)$ are active in xy, see Figure 4: The western edge of the vertex $b_{W,1}$ is active by definition, and since xy satisfies Φ and $\mathtt{PASS}$ at $b_{W,1}$, this vertex must be in state or , so its eastern edge is also active. The same follows inductively for all vertices in the row $(W, \star)$. By the same argument, rotated about 90 degrees, all northern and southern edges of vertices in row $(\star, N)$ are active in xy.

By a similar argument, no other edges are active, and we conclude that y is uniquely determined by x. Furthermore, if E and S denote the active indices in x_E and x_S, then we observe that $W = E$ and $N = S$, since otherwise xy could not satisfy $b_{W,n}$ and $b_{n,N}$. Hence, xy satisfies Φ only if $\varphi_{prop}(x)$ holds. We obtain

$$\mathrm{Sig}(\Phi, x) = 0 \qquad \text{if } \neg\varphi_{prop}(x).$$

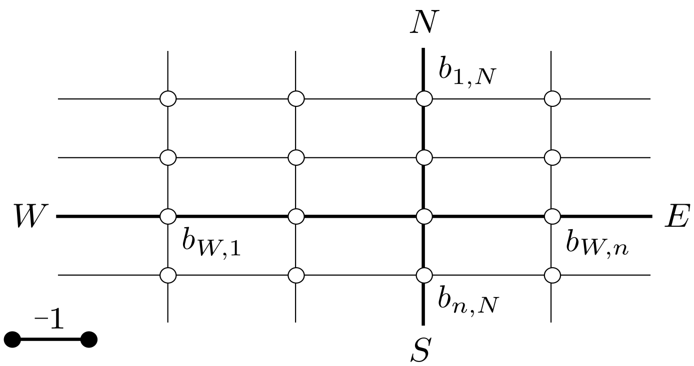

Figure 4: The unique assignment y to $E(\Phi)$ that extends x. Active edges are drawn with thicker lines than non-active edges. Note that the edge of weight -1 with $\texttt{HW}_{=1}$ at its endpoints must be active in any satisfying assignment.

If $\varphi_{prop}(x)$ holds, then $b_{W,N}$ is in state under xy, while the $n-1$ other vertices in row $(W,\star)$ are in state , the $n-1$ other vertices in column $(\star,N)$ are in state , and the remaining n^2-2n+1 vertices are in state . Furthermore, we have the additional active edge of weight -1. Hence, in conclusion, $\varphi_{prop}(x)$ implies

$$\begin{aligned} \mathrm{Sig}(\Phi,x) &= \mathrm{val}(\Phi,xy) \\ &= (-1)\cdot \texttt{PASS}(\;)\cdot \texttt{PASS}(\;)^{n-1}\cdot \texttt{PASS}(\;)^{n-1}\cdot \texttt{PASS}(\;)^{n^2-2n+1} \\ &= 1. \end{aligned}$$

This proves (8).

5.3.2 Calculating the signature of $\Phi'(A)$

Let $\Phi' = \Phi'(A)$ for some fixed $A \subseteq [n]^2$, let $D \subseteq E(\Phi')$ denote the dangling edges of Φ' and let $F = I(a_1)\cup I(a_2)$ denote the set of edges incident with either of the apices a_1 or a_2 in Φ'. Let

$$x \in \{0,1\}^{4n}$$

be an assignment to D that satisfies the predicate $\varphi_{one}(x)$, and let $xyz \in \{0,1\}^{E(\Phi')}$ be a satisfying assignment to the edges of Φ' that extends x, with

$$\begin{aligned} y &\in \{0,1\}^{E(\Phi')\setminus(F\cup D)}, \\ z &\in \{0,1\}^{F}. \end{aligned}$$

We consider the restriction of xyz to xy, that is, to edges not incident with any apex. By definition of $\texttt{PASS}$ and $\texttt{ACT}$, we have, for every vertex $b \in V(\Phi')\setminus\{a_1,a_2\}$, that

$$(xy)|_{I(b)} \in \{\;,\;,\;,\;\}. \tag{12}$$

Recall from Remark 5.1 that we decompose x into x_N, x_E, x_S, x_W, and write $W \in [n]$ and $N \in [n]$ for the unique non-zero index in $x_W \in \{0,1\}^n$ and $x_N \in \{0,1\}^n$, respectively. Since $(xy)|_{I(b)} \in \mathrm{supp}(\texttt{PASS})$ holds by (12) and the definition of $\texttt{PASS}$, the same argument as in the previous subsection for Φ shows that the western and eastern edges of all vertices in row $(W,\star)$ are active under xy, as well as the northern and southern edges of all vertices in the column $(\star,N)$. Likewise, as seen in the previous subsection, it shows that no other edges in $E(\Phi')\setminus F$ are active, that y is unique if $\varphi_{prop}(x)$ holds, and that y does not exist otherwise. This last statement implies that

$$\mathrm{Sig}(\Phi',x) = 0 \qquad \text{if } \neg\varphi_{prop}(x).$$

In the following, let $x \in \{0,1\}^D$ be an assignment to the dangling edges of Φ' that satisfies $\varphi_{prop}(x)$, and let $xy \in \{0,1\}^{E(\Phi')\setminus F}$ be its unique extension to edges not incident with apices, as seen for Φ. We consider the possible assignments $z \in \{0,1\}^F$ to the apex edges such that xyz satisfies Φ'. Here, while the choice of y was unique, the choice of z is not unique.

By virtue of $\mathtt{HW}_{=1}$ at the apex vertices a_1 and a_2, there are unique indices $\tau, \tau' \in A$ such that the edges $a_1 b_\tau$ and $a_2 b_{\tau'}$ are active in xy. By definition of ACT, we actually have $\tau = \tau'$, since all elements in supp(ACT) end on 00 or 11. We write $\tau^* := \tau = \tau'$ for the unique "apex-matched" index, and $b^* := b_{\tau^*}$ for the unique "apex-matched" vertex. By definition of ACT, we have

$$(xyz)|_{I(b^*)} \in \{ ϕ\,11, ┿11\}.$$

It follows that the second component of τ^* must be equal to N, since only vertices in $(\star, N)$ have state ϕ or ┿ under xy. There are T vertices with signature ACT in row $(\star, N)$, by the balance property of our instance $\mathcal{T}$ to GridTiling, and we can choose any of these vertices to be apex-matched. To determine the set of such possible choices, we distinguish two cases, depending on whether $(W, N) \in A$ or not.

$(W,N) \notin A$: The apex-matched vertex must be in state ϕ 11 under xyz. It cannot be in state ┿11, since only $b_{W,N}$ can have state ┿ among its first four edges, but $b_{W,N}$ has PASS assigned, since $(W,N) \notin A$. This gives T assignments z such that xyz satisfies Φ'. Each of the T assignments xyz satisfies $\mathrm{val}_{\Phi'}(xyz) = -1$, because there is (i) one vertex in state ┿00, which contributes a factor of -1 to $\mathrm{val}_{\Phi'}(xyz)$, and (ii) some number of vertices in states • 00, ϕ 00 and -•-00, which however all contribute a unit factor to $\mathrm{val}_{\Phi'}(xyz)$. This implies that $\mathrm{Sig}(\Phi', x) = -T$ if both $(W,N) \notin A$ and $\varphi_{prop}(x)$ hold.

$(W,N) \in A$: The apex-matched vertex may be in state ϕ 11 or ┿11. We make a distinction into these two individual sub-cases:

ϕ 11: We proceed as in the case of $(W,N) \notin A$, but we have only $T-1$ choices left for the apex-matched vertex, since $b_{W,N}$ must have state ┿ among its first four edges and can thus not be in state ϕ 11. This gives $T-1$ assignments z with $\mathrm{val}_{\Phi'}(xyz) = \mathtt{PASS}(┿) = -1$ for each z. (In the expression of $\mathrm{val}_{\Phi'}(xyz)$, we ignored the vertices in states • 00, ϕ 00 and -•-00 that contribute a unit factor.)

┿11 : Since only $b_{W,N}$ can have state ┿ among its first four edges, the apex-matched vertex must be $b_{W,N}$. This gives one assignment z, and $\mathrm{val}_{\Phi'}(xyz) = \mathtt{ACT}(┿) = 1$. Again, we ignored unit factors.

In total, if both $(W,N) \in A$ and $\varphi_{prop}(x)$ hold, then we obtain

$$\mathrm{Sig}(\Phi', xyz) = (T-1)\cdot(-1) + 1 = -T + 2$$

This proves (9), and thus Lemma 5.5. The proof of Theorem 1.2 is completed.

6 The permanent modulo 2^k

We prove Theorem 1.3, which asserts $\oplus$W[1]-hardness of evaluating the permanent mod 2^k. We reduce from the problem $\oplus$GridTiling, the parity version of GridTiling from Definition 2.4. From a high level, the proof resembles that of Theorem 1.2, but the setting of modular evaluation requires us to apply linearly combined signatures in a more intricate way.

6.1 The main idea

Our reduction is based upon the following observation: Let $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ be an instance for $\oplus$GridTiling. For $\omega : \mathcal{C} \to [2]$, recall the graphs H_ω and the numbers $d(\omega)$ from the last section. We can rewrite (11) as

$$2^{|\mathcal{C}|} \cdot \#\mathrm{GridTiling}(\mathcal{T}) = \sum_{\omega:\mathcal{C}\to[2]} T^{d(\omega)} \cdot \mathrm{perm}(H_\omega). \tag{13}$$

Theorem 2.9 asserts that computing $\oplus$GridTiling($\mathcal{T}$) is $\oplus$W[1]-hard. Let $M := 2^{|\mathcal{C}|}$ and assume we could evaluate $\mathrm{perm}(H_\omega)$ modulo $2M$ for all ω. Using arithmetic in $\mathbb{Z}/2M\mathbb{Z}$, we could then evaluate the entire right-hand-side of (13), and this allows us to compute

$$M \cdot \#\mathrm{GridTiling}(\mathcal{T}) \equiv_{2M} \begin{cases} M & \text{if } \#\mathrm{GridTiling}(\mathcal{T}) \text{ is odd,} \\ 0 & \text{if } \#\mathrm{GridTiling}(\mathcal{T}) \text{ is even.} \end{cases}$$

Hence, it seems that we could solve $\oplus$GridTiling($\mathcal{T}$) with an oracle for the permanent modulo $2M = 2^{|\mathcal{C}|+1}$, and we might be tempted to believe that we just proved Theorem 1.3.

However, the above argument suffers from a fatal gap: The graphs H_ω from the previous section feature edges of weight $\frac{1}{2}$, a number that does not exist in the rings $\mathbb{Z}/2^k\mathbb{Z}$ for $k \in \mathbb{N}$. In other words, the proof fails for the surprisingly philosophical reason that the instances H_ω constructed in the previous section do not even *exist* modulo 2^k. More precisely, it is the matchgate $\Gamma_{\mathtt{ACT}}$ used to realize the signature `ACT` that features this offending weight, and it is incurred by the part that we called the *even filter*. To obtain graphs H_ω that avoid edge-weights with even denominators, we therefore construct cell gates using the signature `PRE` rather than its more benign version `ACT`. This adds several complications to our arguments, which we can however handle with a suitable linear combination.

6.2 Revisiting the cell gate

Let $A \subseteq [n]^2$ be fixed in the following, and recall the gates Φ and Φ' from Definition 5.4. Note that Φ features only occurrences of `PASS`, which is realized by the matchgate $\Gamma_{\mathtt{PASS}}$ on edge-weights -1 and 1. We can therefore also realize this gate modulo 2^k. This does not apply to the gate $\Phi'(A)$, as the matchgate $\Gamma_{\mathtt{ACT}}$ realizing `ACT` features the weight $\frac{1}{2}$. We modify $\Phi'(A)$ to a new gate $\Gamma(A)$ by replacing all occurrences of `ACT` with `PRE`.

Definition 6.1. For $A \subseteq [n]^2$, let the gate $\Gamma(A)$ on $4n$ dangling edges be defined exactly as the gate $\Phi'(A)$ from Definition 5.4, but replace every occurrence of `ACT` by `PRE`.

For all $u, v \in [n]$, let $\alpha_{u,v}$ denote the number of occurrences of `PRE` among vertices b_τ with $\tau \in \{(1, v), \ldots, (u-1, v)\}$. Likewise, let $\beta_{u,v}$ denote the number of occurrences of `PRE` among vertices b_τ with $\tau \in \{(u+1, v), \ldots, (n, v)\}$.

Figuratively speaking, $\alpha_{u,v}$ is the number of occurrences of `PRE` in the column above (u, v), and $\beta_{u,v}$ is the number of occurrences below it. In Section 5.2, we used the vertical balance property to ensure that $\alpha_{u,v} + \beta_{u,v}$ is equal to $T-1$ when $(u, v) \in A$, and equal to T when $(u, v) \notin A$. In this section, this vertical balance will not be required, but *horizontal* balance will prove useful instead, for different reasons. For the remainder of our proofs, we define the following auxiliary polynomials, for all $u, v, w \in [n]$:

$$q_u := \sum_{z\in[n]} \alpha_{u,z} \cdot \beta_{u,z} - \binom{\alpha_{u,z}}{2} - \binom{\beta_{u,z}}{2}, \tag{14}$$

$$p_{u,v,w} := (\alpha_{u,v} - \beta_{u,v}) \cdot (\beta_{u,w} - \alpha_{u,w}), \tag{15}$$

$$r_{u,v} := \sum_{\substack{z\in[n]\setminus\{v\} \\ (u,z)\in A}} \beta_{u,z}, \tag{16}$$

$$s_{u,v} := \sum_{\substack{z\in[n]\setminus\{v\} \\ (u,z)\in A}} \alpha_{u,z}. \tag{17}$$

Using these polynomials, we can express the signature of Γ.

Lemma 6.2. *Let $A \subseteq [n]^2$, let $\Gamma = \Gamma(A)$ and let $x \in \{0,1\}^{4n}$ satisfy φ_{one}. Recall the conventions from Remark 5.1, including that we implicitly decompose the string x into x_N, x_E, x_S, x_W.*

- *If $x_W \neq x_E$ or $\mathrm{hw}(x_S) \neq 1$, then $\mathrm{Sig}(\Gamma, x) = 0$.*
- *If $\varphi_{prop}(x)$ is true (i.e., we have $x_W = x_E$ and additionally $x_N = x_S$), write $u := x_W$ and $v := x_N$, with $u, v \in [n]$. Note that these numbers are well-defined. We call such assignments x* wanted*, and we have*
$$\mathrm{Sig}(\Gamma, x) = \begin{cases} q_u - r_{u,v} - s_{u,v} - \alpha_{u,v} - \beta_{u,v} & \text{if } (u,v) \notin A \\ q_u - r_{u,v} - s_{u,v} + 1 & \text{if } (u,v) \in A \end{cases}$$
- *If $\varphi_{prop}(x)$ is false (i.e, we have $x_W = x_E$, but $x_N \neq x_S$), then write $u := x_W$, $v := x_N$, and $w := x_S$. We call such assignments x* unwanted*, and we have*
$$\mathrm{Sig}(\Gamma, x) = \begin{cases} p_{u,v,w} & \text{if } (u,v) \notin A,\ (u,w) \notin A \\ p_{u,v,w} + \alpha_{u,v} - \beta_{u,v} & \text{if } (u,v) \notin A,\ (u,w) \in A \\ p_{u,v,w} + \beta_{u,w} - \alpha_{u,w} & \text{if } (u,v) \in A,\ (u,w) \notin A \\ p_{u,v,w} + \beta_{u,w} - \alpha_{u,w} + \alpha_{u,v} - \beta_{u,v} + 1 & \text{if } (u,v) \in A,\ (u,w) \in A \end{cases}$$

The full proof of this lemma requires a somewhat tedious calculation, which is deferred to Section 6.4. Note that the entries of $\mathrm{Sig}(\Gamma)$ are polynomials in the indeterminates $\alpha_{u,v}$ and $\beta_{u,v}$ for $u, v \in [n]$

Taking Lemma 6.2 for granted at the moment, we note that the gate Γ essentially discriminates between six different assignment types, depending on whether x is wanted (giving 2 types) or unwanted (giving 4 types, depending on whether (x_W, x_N) and (x_W, x_S) are each contained in A). However, the actual value of $\mathrm{Sig}(\Gamma, x)$ is *not constant* for each of the six types, as it depends on u, v, w and the concrete values for $\alpha_{u',v'}$ and $\beta_{u',v'}$ for all $u', v' \in [n]$. Compare this to the gate Φ' from Section 5.3.2, which attains one of the three fixed values $\{0, -T, -T+2\}$ due to vertical balance. It turns out that the simple balance argument used in the last section does not work in this setting; our technical efforts in the remainder of the proof therefore aim at the following two goals:

Goal 1: Ensure that the four unwanted cases (as defined above) cancel out.

Goal 2: Ensure that the two wanted cases (as defined above) do not depend upon the actual value of (x_W, x_N), but only on the information whether $(x_W, x_N) \in A$ or $(x_W, x_N) \notin A$.

In the following, we show how to attain these goals by considering a particular linear combination of matchgate signatures that could be considered as the "derivative" of a matchgate.

6.3 Linear combinations via discrete derivatives

Recall the construction of Γ from Definition 6.1. In the following, we construct a gate $\Gamma_\uparrow$ from Γ by adding several "dummy" vertices. Then we consider the difference
$$\mathrm{Sig}(\Gamma_\uparrow) - \mathrm{Sig}(\Gamma).$$
The gate $\Gamma_\uparrow$ is obtained from Γ by adding dummy rows of vertices with signature `PRE`, and this allows us to obtain $\mathrm{Sig}(\Gamma_\uparrow)$ by a simple substitution on the indeterminates of $\mathrm{Sig}(\Gamma)$.

Definition 6.3. We define a *dummy gate* as in Figure 5: Starting from a vertex with signature `PRE`, add several vertices of signature $\mathtt{HW}_{=1}$ to its western and eastern dangling edges to force these edges to be inactive, as shown in the left part of the figure. We then define a *dummy row* by arranging n dummy gates horizontally as shown in the right part of the figure.

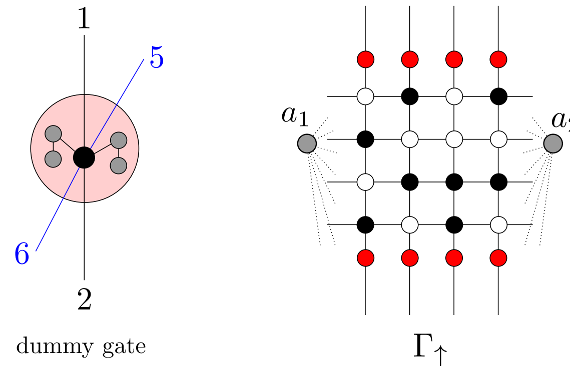

Figure 5: A dummy gate is shown on the left. On the right, we see $\Gamma_\uparrow$, which is obtained from Γ by adding rows of dummy gates, shown red. Each gray vertex is assigned $\mathtt{HW}_{=1}$, and the apices connect to all black vertices (assigned `PRE`) and all red vertices (whose signature is realized by the dummy gate). White vertices are assigned `PASS`, and they are not adjacent to apices.

Starting from Γ, define a gate $\Gamma_\uparrow$ by adding a dummy row above the row $(1, \star)$, and a dummy row below the row $(n, \star)$, as shown in Figure 5. We connect apex a_1 to the dangling edge 5 of each dummy gate, and a_2 to the dangling edge 6.

Furthermore, we define algebraic manipulations on multivariate polynomials that capture the effect of introducing dummy rows into Γ as described above.

Definition 6.4. Let p be any multivariate polynomial on the indeterminates $\alpha_{u,v}$ and $\beta_{u,v}$ for $u, v \in [n]$. Write $x \leftarrow y$ for the operation of substituting x with y in p. Then we define $p_\uparrow$ to be the polynomial obtained from p after performing the substitutions $\alpha_{u,v} \leftarrow \alpha_{u,v} + 1$ and $\beta_{u,v} \leftarrow \beta_{u,v} + 1$ for all $u, v \in [n]$.

We also define the following discrete derivative operator D on such polynomials p:

$$D(p) := p_\uparrow - p.$$

The following is then easily observed:

Lemma 6.5. *We have*

$$\mathrm{Sig}(\Gamma_\uparrow) = (\mathrm{Sig}(\Gamma))_\uparrow,$$

and in particular, we have

$$D(\mathrm{Sig}(\Gamma)) = \mathrm{Sig}(\Gamma_\uparrow) - \mathrm{Sig}(\Gamma).$$

Note that the operator D indeed resembles a derivative: We have linearity by $D(p+q) = D(p) + D(q)$, and applying D to a polynomial p of degree d gives one of degree $d-1$. We will use these properties of D to effect two useful modifications on the polynomials in (14)-(16), and thus ultimately on $\mathrm{Sig}(\Gamma)$. These correspond to the two goals described at the end of Section 6.2.

1. Concerning the first goal, our choice of D ensures that "unwanted" polynomials vanish under D. For instance, for all $u, v, w \in [n]$, the polynomial $p_{u,v,w}$ from (15) maps to

$$\begin{aligned} D(p_{u,v,w}) &= ((\alpha_{u,v}+1) - (\beta_{u,v}+1)) \cdot ((\beta_{u,w}+1) - (\alpha_{u,w}+1)) \\ &\quad -(\alpha_{u,v} - \beta_{u,v}) \cdot (\beta_{u,w} - \alpha_{u,w}) \\ &= 0. \end{aligned} \tag{18}$$

By our calculation of $\mathrm{Sig}(\Gamma)$, this implies that $D(\mathrm{Sig}(\Gamma))$ vanishes on assignments x with $x_N \neq x_S$ and $(x_W, x_N) \notin A$ and $(x_W, x_S) \notin A$. The other unwanted cases will be handled by similar arguments.

2. Under the operator D, linear terms, such as $\alpha_{u,v}$ or $\beta_{u,v}$ for $u, v \in [n]$, are mapped to

$$\begin{aligned} D(\alpha_{u,v}) &= (\alpha_{u,v}+1) - \alpha_{u,v} \;=\; 1, && (19) \\ D(\beta_{u,v}) &= (\beta_{u,v}+1) - \beta_{u,v} \;=\; 1. && (20) \end{aligned}$$

   This helps us to attain the second goal, since the original terms depend on the concrete values of $\alpha_{u,v}$ or $\beta_{u,v}$ in A, whereas the resulting constants do not. It will also turn out that only linear terms need to be considered.

In the following, we show that $D(\mathrm{Sig}(\Gamma))$ essentially realizes the function g_κ, up to some additive term on assignments x with φ_{prop}. This allows us to write g_κ as a linear combination of the matchgate signatures $\mathrm{Sig}(\Gamma_\uparrow)$ and $\mathrm{Sig}(\Gamma)$. As a technical requirement, we use Lemma 2.10 to ensure that the set A in the definition of $\Gamma = \Gamma(A)$ is horizontally balanced.

Lemma 6.6. *Assume the existence of a number $T \in \mathbb{N}$ such that A features exactly T elements of type $(u, \star)$, for all $u \in [n]$. Let $\Gamma = \Gamma(A)$ and write $D := D(\mathrm{Sig}(\Gamma)) = \mathrm{Sig}(\Gamma_\uparrow) - \mathrm{Sig}(\Gamma)$. We then have*

$$D = \begin{cases} 0 & \text{if } \neg\varphi_{prop}(x) \\ \begin{cases} n - 2T - 2 & (x_W, x_N) \notin A \\ n - 2T + 2 & (x_W, x_N) \in A \end{cases} & \text{if } \varphi_{prop}(x) \end{cases}$$

Proof. We prove the identity using linearity of D. For all $u, v, w \in [n]$, consider the effect of D on the polynomials from (14)-(17). For instance, we have seen in (18) and (19)-(20) that

$$\begin{aligned} D(p_{u,v,w}) &= 0, \\ D(\alpha_{u,v}) = D(\beta_{u,v}) &= 1. \end{aligned}$$

Likewise, we can show that

$$\begin{aligned} D(q_u) &= \sum_{v \in [n]} 1 \;=\; n, \\ D(r_{u,v}) = D(s_{u,v}) &= \sum_{\substack{z \in [n]\setminus\{v\} \\ (u,z) \in A}} 1 \;=\; \begin{cases} T & (u,v) \notin A, \\ T - 1 & (u,v) \in A. \end{cases} \end{aligned}$$

Together with linearity of D and the expression of $\mathrm{Sig}(\Gamma)$ from 6.2, this proves the claim by a simple calculation for each of the six assignment types. ∎

Corollary 6.7. *Write $S := n - 2T - 2$ and recall the matchgate Φ from Section 5.3.1 with*

$$\mathrm{Sig}(\Phi, x) = \begin{cases} 1 & \text{if } \varphi_{prop}(x), \\ 0 & \text{otherwise.} \end{cases}$$

Then the following linear combination realizes the signature g_κ:

$$\frac{D - S \cdot \mathrm{Sig}(\Phi)}{4} = \frac{\mathrm{Sig}(\Gamma_\uparrow) - \mathrm{Sig}(\Gamma) - S \cdot \mathrm{Sig}(\Phi)}{4}.$$

Note that each of the constituent gates $\Gamma_\uparrow$, Γ and Φ has at most two apices and features only edge-weights from the set $\{-1, 1\}$. Furthermore, each of these gates admits a 2-coloring.

Using Corollary 6.7, we can complete the proof of Theorem 1.3. Recall that we aim at a reduction from $\oplus$GridTiling to the permanent modulo 2^k.

Proof of Theorem 1.3. Let $\mathcal{A} = (n, k, \mathcal{C}, \mathcal{T})$ be an instance for the $\oplus\mathsf{W}[1]$-complete problem $\oplus$GridTiling. To prove the lower bound under $\oplus\mathsf{ETH}$, we may assume $|\mathcal{C}| = \mathcal{O}(k)$ by Theorem 2.9. Furthermore, by horizontal balance via Lemma 2.10, we may assume that we are given a number $T \in \mathbb{N}$ such that $|\mathcal{T}(\kappa) \cap (u, \star)| = T$ for all $\kappa \in \mathcal{C}$ and $u \in [n]$.

Recall Definition 5.2 and Lemma 5.3 of Section 5.1: These allow us to compute a signature graph G with signatures f_κ at $\kappa \in [k]^2 \setminus \mathcal{C}$ and signatures g_κ at $\kappa \in \mathcal{C}$ such that

$$\#\mathrm{GridTiling}(\mathcal{A}) = \mathrm{Holant}(G).$$

As shown in Lemma 5.5, we can realize f_κ by the planar matchgate Φ on edge-weights $\{-1, 1\}$. Furthermore, as shown in Lemma 6.6, we can realize g_κ for each $\kappa \in \mathcal{C}$ as the linear combination of three 2-apex matchgates on edge-weights $\{-1, 1\}$: Let $\Gamma_\kappa := \Gamma(\mathcal{T}(\kappa))$ be as in Definition 6.1, and let $\Gamma_{\kappa,\uparrow}$ be obtained from Γ_κ as in Definition 6.3. Then, similarly to the proof of Theorem 1.2, we obtain with Lemma 6.6 and Lemma 3.8 about the linear combinations of signatures that

$$4^{|\mathcal{C}|} \cdot \mathrm{Holant}(G) = \sum_{\omega:\mathcal{C}\to[3]} (-1)^{d(\omega)} \cdot (-S)^{e(\omega)} \cdot \mathrm{PerfMatch}(H_\omega). \tag{21}$$

Here, for each $\omega : \mathcal{C} \to [3]$, the number $d(\omega)$ is defined to be the number of 2-entries in ω, and $e(\omega)$ is the number of 3-entries. The graph H_ω is obtained as follows: For $\kappa \in [k]^2 \setminus \mathcal{C}$, insert the matchgate Φ at the cell vertex c_κ. For all $\kappa \in \mathcal{C}$, insert $\Gamma_{\kappa,\uparrow}$ or Γ_κ or Φ at c_κ if $\omega(\kappa)$ is 1 or 2 or 3, respectively.

Let $M := 2^{2|\mathcal{C}|}$. With an oracle for computing $\mathrm{PerfMatch}(H_\omega)$ modulo $2M$ for all ω, we can compute the right-hand side of (21) modulo $2M$ via arithmetic in $\mathbb{Z}/2M\mathbb{Z}$. We then obtain the value (modulo $2M$) of

$$M \cdot \mathrm{Holant}(G) = M \cdot \#\mathrm{GridTiling}(\mathcal{A}) \equiv_{2M} \begin{cases} M & \text{if } \#\mathrm{GridTiling}(\mathcal{A}) \text{ odd,} \\ 0 & \text{if } \#\mathrm{GridTiling}(\mathcal{A}) \text{ even.} \end{cases}$$

Each graph H_ω is bipartite, has at most $2|\mathcal{C}| = \mathcal{O}(k)$ apices, and the computation is modulo $2M = 2^{\mathcal{O}(k)}$. We have thus shown a parameterized Turing reduction from $\oplus$GridTiling to the evaluation of the permanent on $\mathcal{O}(k)$-apex graphs modulo $2^{\mathcal{O}(k)}$. Since Theorem 2.9 asserts the $\oplus\mathsf{W}[1]$-completeness of the former problem, the theorem follows. $\square$

6.4 Calculating the signature of Γ

In the remainder of this section, we prove Lemma 6.2. Let $x \in \{0, 1\}^{4n}$ be an assignment to the dangling edges of Γ. The statement of the lemma is shown by inspecting the possible satisfying extensions of x, as we did when calculating $\mathrm{Sig}(\Phi')$. To understand the following proof, we therefore recommend recalling Section 5.3.2, since that section contains a similar, yet substantially simpler argument.

Let $F \subseteq E(\Gamma)$ denote the edges of Γ that are incident with apices. Given x, let $xyz \in \{0, 1\}^{E(\Gamma)}$ be an assignment extending x such that $\mathrm{Sig}(\Gamma, xyz) \neq 0$, with $y \in \{0, 1\}^{E(\Gamma)\setminus F}$ and $z \in \{0, 1\}^F$. Due to $\mathtt{HW}_{=1}$ at the apex vertices a_1 and a_2 of Γ, there are apex-matched indices $\tau_1, \tau_2 \in A$ and apex-matched vertices $b_1 := b_{\tau_1}$ and $b_2 := b_{\tau_2}$ such that a_1b_1 and a_2b_2 are active in xyz. However, opposing Section 5.3.2, it may well be that $\tau_1 \neq \tau_2$, and this makes our calculations somewhat more difficult. In particular, the assignment y is no longer uniquely determined by x.

For each assignment x, we partition the satisfying extending assignments xyz to Γ into six partition classes $\{\mathcal{P}_i(x)\}_{i\in[6]}$, corresponding to the states of the (at most two distinct) apex-matched vertices. More precisely, for $i \in [6]$, we let

$$\mathcal{P}_i(x) := \{xyz \in \{0, 1\}^{E(\Gamma)} \mid xyz|_{I(b_1)} \text{ and } xyz|_{I(b_2)} \text{ are as in row } i \text{ of Table 1}\}.$$

Note that b_1 and b_2 depend upon the assignment xyz. To give an example, in row 1, and thus in class $\mathcal{P}_1$, we consider extending assignments xyz that have only one vertex with active edges leading to an apex, and the local assignment at this vertex reads ╋11. More formally, we have

$$b_1 = b_2 \ \wedge \ xyz|_{I(b_1)} = \text{╋}11.$$

	$(u,v) \notin A$	$(u,v) \in A$	$(u,v)\notin A$ $(u,w)\notin A$	$(u,v)\notin A$ $(u,w)\in A$	$(u,v)\in A$ $(u,w)\notin A$	$(u,v)\in A$ $(u,w)\in A$
11	0	1	0	0	0	0
11	$-\alpha_{u,v} - \beta_{u,v}$	$-\alpha_{u,v} - \beta_{u,v}$	0	0	0	0
10, 01	q_u	q_u	$p_{u,v,w}$	$p_{u,v,w}$	$p_{u,v,w}$	$p_{u,v,w}$
10, 01	$-r_{u,v}$	$-r_{u,v} + \alpha_{u,v}$	0	$\alpha_{u,v} - \beta_{u,v}$	0	$\alpha_{u,v} - \beta_{u,v}$
10, 01	$-s_{u,v}$	$-s_{u,v} + \beta_{u,v}$	0	0	$\beta_{u,w} - \alpha_{u,w}$	$\beta_{u,w} - \alpha_{u,w}$
10, 01	0	0	0	0	0	1

Table 1: The six assignment types of the cell are listed as columns, and the possible states of the (at most two) apex-matched vertices are listed as rows. The signature of Γ on each of the six assignment types is given as the sum of the elements in the corresponding column. Note that the table is divided into four quadrants. We have essentially already calculated the top left quadrant in Section 5.3 when we calculated $\mathrm{Sig}(\Phi')$.

As another example, in row 3, we have

$$b_1 \neq b_2 \ \wedge\ xyz|_{I(b_1)} = \ 10 \ \wedge\ xyz|_{I(b_2)} = \ 01.$$

It is evident that, given $x \in \{0,1\}^{4n}$, we have

$$\mathrm{Sig}(\Gamma, x) = \sum_{i\in[6]} \underbrace{\sum_{xyz\in\mathcal{P}_i(x)} \mathrm{val}_\Gamma(xyz)}_{=:P_i(x)}. \tag{22}$$

In Table 1, we calculate $P_i(x)$ for all $i \in [6]$ and all six types of assignments x to dangling edges distinguished by the signature: The entry in this table at row $i \in [6]$ and column $j \in [6]$ denotes the number $P_i(x)$ on assignments x of the j-th type. Note that the table is divided into four quadrants, as indicated by the double lines in Table 1. In Section 5.3.2, we have essentially already calculated the values in the top left quadrant. In the following, we calculate the remaining quadrants.

Before doing so, we first need to make some general observations: In each satisfying assignment xyz extending x, all western and eastern edges of vertices in the row $(x_W, \star)$ are active, and no other western and eastern edges are active. This is because for any vertex $b \in V(\Gamma) \setminus \{a_1, a_2\}$, the signatures `PASS` and `PRE` imply that the assignment $xy|_{I(b)}$ has one of the states

$$\underbrace{\ ,\ ,\ ,\ }_{\text{“tame”}},\underbrace{\ ,\ ,\ ,\ }_{\text{“wild”}}. \tag{23}$$

In each such state, be it tame or wild, the western incident edge is active iff the eastern edge is active as well. By an argument as in Section 5.3.1, this implies $x_W = x_E$ for the assignment x. Note that a similar statement from north to south is not necessarily true, as witnessed by vertices in a “wild” state.

If $b_1 \neq b_2$, this implies $xyz|_{I(b_1)} \in \{ 10, 10\}$ and $xyz|_{I(b_2)} = \{ 01, 01\}$. Because all other vertices are in tame states and thus enforce equality on their northern and southern dangling edges, the vertex b_1 “shoots” a ray of active vertical edges to the north (transmitted by vertices in state , , 00, 00). This ray may either leave the cell, or it hits b_2. We conclude that, for any column $j \in [n]$,

- $x_N(j) = x_S(j)$ iff column $(\star, j)$ contains neither b_1 nor b_2, or it contains both,
- $x_N(j) = 1 \ \wedge\ x_S(j) = 0$ iff column $(\star, j)$ contains b_1 but not b_2,
- $x_N(j) = 0 \ \wedge\ x_S(j) = 1$ iff column $(\star, j)$ contains b_2 but not b_1.

We are now ready to calculate the remaining quadrants of Table 1. Recall that we use the abbreviations $u := x_W$, $v := x_N$ and $w := x_S$.

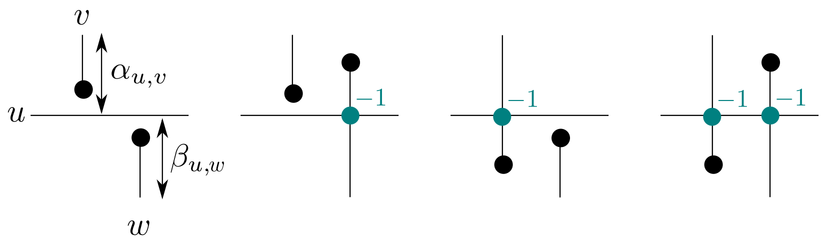

Figure 6: Relevant states in the bottom right quadrant. The vertices b_1 and b_2 are shown as black dots, crossings with the horizontal path are shown as turquoise dots.

Top right quadrant: If $x_N \neq x_S$, then $\mathcal{P}_1(x) = \mathcal{P}_2(x) = \emptyset$. This is because all vertices in assignments $xyz \in \mathcal{P}_1(x) \cup \mathcal{P}_2(x)$ are in tame states, which would imply $x_N = x_S$. This explains all zeros in the top right quadrant of Table 1.

Bottom right quadrant (0/1 entries): If $x_N \neq x_S$ and $(x_W, x_N) \notin A$, then no satisfying assignment has a vertex in state ┴: By our general observation, the index of this vertex would be (x_W, x_N), but this vertex has no adjacent apex, since $(x_W, x_N) \notin A$, and it can thus only be in a tame state. Likewise, if $(x_W, x_S) \notin A$, then no satisfying assignment has a vertex in state ┬. This explains all zeros in the bottom right quadrant of Table 1, and it also explains the bottom right entry of 1.

Bottom right quadrant (other entries): By our general observation, the vertex b_1 must be located in the column $(\star, v)$ and b_2 must be located in the column $(\star, w)$.

Consider the third row in the right quadrant and Figure 6. Because of the states of b_1 and b_2, neither of them is on the horizontal path u. This gives $\alpha_{u,v} + \beta_{u,v}$ choices for b_1. When b_1 is above (u, v), there are $\alpha_{u,v}$ possibilities, and the northbound ray emitted by b_1 does not cross the horizontal path in $(u, \star)$ described in the general observations. When b_1 is below (u, v), there are $\beta_{u,v}$ possibilities, and the northbound ray crosses the horizontal path in $(u, \star)$, so the vertex at (u, v) contributes a factor -1 from `PASS`(┼) or `PRE`(┼00). By a similar analysis for b_2 as for b_1, we obtain four cases, shown in Figure 6 and we see that, for inputs x of the third type in Table 1, we have

$$
\begin{aligned}
P_3(x) &= \alpha_{u,v} \cdot \beta_{u,w} - \alpha_{u,v} \cdot \alpha_{u,w} - \beta_{u,v} \cdot \beta_{u,w} + \beta_{u,v} \cdot \alpha_{u,w} \\
&= (\alpha_{u,v} - \beta_{u,v}) \cdot (\beta_{u,w} - \alpha_{u,w}) \\
&= p_{u,v,w}
\end{aligned}
$$

The calculation of the remaining rows of Table 1 is similar, except that b_1 or b_2 may appear on the horizontal path $(u, \star)$ by the ┴ or ┬ state, so only one or fewer factors of $p_{u,v,w}$ remain.

Bottom left quadrant (zero entries): The argument for the zero entries in the bottom right quadrant applies here as well.

Bottom left quadrant (other entries): It can be verified that b_1 and b_2 must be located in the same column, as otherwise it would be impossible to have $x_N = x_S$. In particular, either they are in some column $(\star, j)$ with $j \neq v$, or they are in the column $(\star, v)$. We calculate the weighted sum over the relevant extensions in Table 2, and then use it to get the bottom left quadrant of Table 1. To verify the completeness of our reasoning, we advise to tick the corresponding cells of the table while reading.

Let us assume first that b_1 and b_2 appear in a column $(\star, j)$ of Γ with $j \neq v$. These situations are covered in columns 1, 2, 4, and 5 of Table 2. Then, after fixing the positions of b_1 and b_2, the unique possible assignment realizing this choice contains the horizontal path $(u, \star)$, a vertical path $(\star, v)$ and a path

states of b_1, b_2	$(u,v)\notin A$ $j\neq v$ $(u,j)\notin A$	$(u,v)\notin A$ $j\neq v$ $(u,j)\in A$	$(u,v)\notin A$ $j=v$	$(u,v)\in A$ $j\neq v$ $(u,j)\notin A$	$(u,v)\in A$ $j\neq v$ $(u,j)\in A$	$(u,v)\in A$ $j=v$
⫰10, ⫯01	t_j	t_j	t_j	t_j	t_j	t_j
⫰10, ┳01	0	$-\beta_{u,j}$	0	0	$-\beta_{u,j}$	$\alpha_{u,v}$
┻10, ⫯01	0	$-\alpha_{u,j}$	0	0	$-\alpha_{u,j}$	$\beta_{u,v}$
┻10, ┳01	0	0	0	0	0	0

Table 2: A detailed table of the bottom left quadrant of Table 1.

connecting b_1 and b_2. The vertex at (u,v) yields the value -1, since it is in state ┼ or ╋00. Whether the vertex at (u,j) also yields -1 depends on whether the line segment b_1b_2 crosses the horizontal path $(u,\star)$.

Consider the first row of Table 2 for columns with $j\neq v$. When b_1 and b_2 are in states ⫰10 and ⫯01 respectively, there are $\alpha_{u,j}\beta_{u,j}$ choices for b_1 and b_2 such that the line segment b_1b_2 crosses the horizontal path (and in this case, we have two crossings, each of which yields a factor -1). There are $\binom{\alpha_{u,j}}{2}+\binom{\beta_{u,j}}{2}$ choices of b_1 and b_2 such that the crossing does not occur (yielding one crossing in total and a factor -1). Hence, the total sum over extensions to x with b_1 and b_2 in states ⫰10 and ⫯01 is equal to

$$t_j=\alpha_{u,j}\beta_{u,j}-\binom{\alpha_{u,j}}{2}-\binom{\beta_{u,j}}{2}.$$

We observe that no extension to x can have the vertices b_1 and b_2 in states ┻10 and ┳01, as these states would force the vertices b_1 and b_2 to appear in different columns of Γ. Hence, the number of extensions in row 4 are all zero. Note also that, in columns 1, 3, and 4, no states other than ⫰10 and ⫯01 can appear: Every other state would require $(u,j)\in A$, since only such vertices can possibly be in wild states.

The calculations so far have settled columns 1 and 4; we now consider column 2. If and only if b_2 is located on (u,j), then the vertices b_1 and b_2 are in states ⫰10, ┳01. Then the vertex b_2 at (u,j) gives PRE(┳01) $=1$, and b_1 gives PRE(⫰10) $=1$. The vertex at (u,v) is in state ┼ or ╋00 and consequently yields -1. We observe that there are $\beta_{u,j}$ choices for b_1. This settles row 2 of column 2. A symmetric argument applies in row 3 of column 2, when the vertices b_1 and b_2 are in states ┻10, ⫯01.

The same argument applies to column 5, since both b_1 and b_2 do not appear in the v-th column of Γ. This settles all columns with $j\neq v$; we will henceforth consider the case $j=v$ as in columns 3 and 6. In these columns, the vertices b_1 and b_2 must be situated in column $(\star,v)$ of Γ. Furthermore, we again have the horizontal path passing through row $(u,\star)$.

Consider row 1, corresponding to states ⫰10, ⫯01. Here, it is irrelevant whether $(u,v)\in A$ or not, since none of b_1 or b_2 can be located at (u,v), as the horizontal path could otherwise not pass through these vertices. There are $\alpha_{u,j}\beta_{u,j}$ possible positions for b_1 and b_2 such that b_1 lies above the horizontal path $(u,\star)$ and b_2 lies below it. In both situations, no crossing occurs. Furthermore, there are $\binom{\alpha_{u,j}}{2}+\binom{\beta_{u,j}}{2}$ possible positions for b_1 and b_2 such that both lie above or both lie below the horizontal path, introducing precisely one crossing with the path. Hence, the weighted sum over extensions is again given by t_j, with $j=v$.

This settles column 3; it remains to consider column 6. Consider its second row. Because b_2 is in state ┳01, it is located at (u,v), and shoots a ray to the south. There are $\alpha_{u,v}$ positions left for b_1 to shoot a ray to the north. Similarly, the third entry is $\beta_{u,v}$. It is important to note here that no crossing occurs, as opposed to, say, column 5.

We have now calculated all entries of the table. If we sum the first 3 columns and the last 3 columns, respectively, we get the bottom left quadrant of Table 1. (Note that each block of 3 columns actually corresponds to n choices for j, so each sum involves n terms.)

Conclusion of the calculation. This explains all entries of Table 1. Given an assignment x having one of the types indicated in the columns of Table 1, the value $\mathrm{Sig}(\Gamma,x)$ is then obtained by summing along the corresponding column as in (22).

Acknowledgement

We wish to thank an anonymous reviewer for extensive and helpful comments on the submitted version.

References

[1] Manindra Agrawal. Determinant versus permanent. In *Proceedings of the 25th International Congress of Mathematicians, ICM 2006*, volume 3, pages 985–997, 2006.

[2] Andreas Björklund, Holger Dell, and Thore Husfeldt. The parity of set systems under random restrictions with applications to exponential time problems. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 231–242, 2015.

[3] Andreas Björklund, Thore Husfeldt, and Isak Lyckberg. Computing the permanent modulo a prime power. Unpublished, and personal communication.

[4] Jin-Yi Cai, Xi Chen, and Dong Li. Quadratic lower bound for permanent vs. determinant in any characteristic. *Computational Complexity*, 19(1):37–56, 2010.

[5] Jin-Yi Cai and Vinay Choudhary. Valiant's holant theorem and matchgate tensors. In *In Proceedings of TAMC 2006: Lecture Notes in Computer Science*, pages 248–261, 2006.

[6] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *CoRR*, abs/1303.6729, 2013.

[7] Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures: extended abstract. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 635–644, 2013.

[8] Jin-Yi Cai and Pinyan Lu. Holographic algorithms: From art to science. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 401–410, New York, NY, USA, 2007. ACM.

[9] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 644–653. IEEE Computer Society, 2008.

[10] Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-SAT: An isolation lemma for k-CNFs. *J. Comput. Syst. Sci.*, 74(3):386–393, 2008.

[11] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized np-hard problems. *Inf. Comput.*, 201(2):216–231, 2005.

[12] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346 – 1367, 2006.

[13] Radu Curticapean. Counting matchings of size k is #W[1]-hard. In *ICALP (1)*, pages 352–363, 2013.

[14] Radu Curticapean. Counting perfect matchings in graphs that exclude a single-crossing minor. *CoRR*, abs/1406.4056, 2014.

[15] Radu Curticapean. *The simple, little and slow things count: on parameterized counting complexity.* PhD thesis, Universität des Saarlandes, Postfach 151141, 66041 Saarbrücken, 2015.

[16] Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 130–139, 2014.

[17] Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102(2):283–305, 1992.

[18] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms*, 10(4):21, 2014.

[19] E. Demaine, M. Hajiaghayi, and D. Thilikos. 1.5-approximation for treewidth of graphs excluding a graph with one crossing as a minor. In *APPROX*, pages 67–80, 2002.

[20] E. Demaine, M. Hajiaghayi, and D. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41(4):245–267, 2005.

[21] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 637–646, 2005.

[22] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Approximation algorithms via structural results for apex-minor-free graphs. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, pages 316–327, 2009.

[23] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[24] Rodney G. Downey and Michael R. Fellows. Parameterized computational feasibility. In *Feasible Mathematics II*. Birkhäuser Boston.

[25] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[26] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, (4):892–922, 2004.

[27] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[28] Fedor V. Fomin, Erik D. Demaine, and Mohammad Taghi Hajiaghayi. Bidimensionality. In *Encyclopedia of Algorithms*. 2015.

[29] Anna Galluccio and Martin Loebl. On the theory of Pfaffian orientations. I. Perfect matchings and permanents. *Electronic Journal of Combinatorics*, 6, 1999.

[30] Heng Guo, Pinyan Lu, and Leslie G. Valiant. The complexity of symmetric Boolean parity Holant problems. *SIAM Journal on Computing*, 42(1):324–356, 2013.

[31] Russel Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[32] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Computer and Sys. Sci.*, 63(4):512–530, 2001.

[33] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993.

[34] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

[35] Pieter W. Kasteleyn. The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209 – 1225, 1961.

[36] Pieter W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

[37] Charles Little. An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs. In *Combinatorial Mathematics*, LNCS, pages 63–72. 1974.

[38] Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.

[39] Dániel Marx. A tight lower bound for planar multiway cut with fixed number of terminals. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 677–688, 2012.

[40] Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, pages 542–553, 2014.

[41] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.

[42] Catherine McCartin. Parameterized counting problems. *Ann. Pure Appl. Logic*, 138(1-3):147–182, 2006.

[43] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009.

[44] Tullio Regge and Riccardo Zecchina. Combinatorial and topological approach to the 3d ising model. *Journal of Physics A: Mathematical and General*, 33(4):741–761, 2000.

[45] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43 – 76, 2003.

[46] Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[47] Simon Straub, Thomas Thierauf, and Fabian Wagner. Counting the number of perfect matchings in K_5-free graphs. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 66–77, 2014.

[48] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6(68):1478–6435, 1961.

[49] Glenn Tesler. Matchings in graphs on non-orientable surfaces. *J. Comb. Theory, Ser. B*, 78(2):198–231, 2000.

[50] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.

[51] Leslie G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979.

[52] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

[53] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[54] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[55] Vijay V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Inf. Comput.*, 80(2):152–164, 1989.